EVENT PREDICTION THROUGH STRUCTURAL INTELLIGENCE IN ONLINE SOCIAL NETWORKS

A Thesis submitted by

Leonard Tan, Masters of Biomedical Engineering (MBioMedE)

For the award of

Doctor of Philosophy

2020

ABSTRACT

The Internet today is a platform of information exchange between real people across the globe. Event prediction is an emerging and highly complex topic of interest which enjoys wide ranging applications in fintech, medical, security, etc. Some of these implementations include time sequenced methods, pattern recognition techniques, multiple instance learning, topic based approach, etc. While they have been adequate at handling predictions of events from past discrete occurances, they fall short of the capability to predict events from a continuous stream of social information exchange.

Furthermore, many of these approaches lack the representative power of describing and tracking events through time. Relational flux and turbulence in Online Social Networks (OSNs) can be defined as the complex evolution of social communication patterns staged over important topic contexts which have the potential to cause abberations of relational states. They play very important roles in determining tasks like recognition, prediction, detection, etc. across applications like recommendation, clustering, community, privacy, security, knowledge representation, etc. For example, an essential research question for Knowledge Representation Learning (KRL) is how to explictly embed implict real-life relational states between entities structured in a Knowledge Graph (KG).

Most current studies today however, do not have the capability to effectively generalize relationships across heterogeneous architectures. Indeed, an important challenge to address is that latent communication patterns in local and global contexts of social opinions cannot be fully captured. Thus, event prediction is challenging for two reasons: its generalized, temporal, evolving nature and drifting contexts. In addition, many current approaches however, lack the capacity of describing and tracking general events over time. To

tackle these issues, this study develops a novel RFT model which leverages on the mechanics of Relational Flux and Turbulence to model dynamic communicative behaviors between actors within social networks. To the best knowledge offered by existing literature, there has not been a similar model and / or method of approach which effectively predicts events from a computationally cognitive perspective.

To surmise the milestones achieved by this research endeavour, extensive experiments were conducted on large-scale datasets from Twitter, Googlefeed and Livejournal. From the experimental results, it was shown that RFT is able to identify and predict relational turbulence in a social flux which mirrors real life relational state transitions in a social topic context. The following demonstration from the F1-scores and k-fold cross validation results proves that the model performs comparably better by more than 10% to well-known predictors such as the Hybrid Probabilistic Markovian (HPM) predictive method [1] and other state-of-the-art baselines in predicting events. Importantly, this research development proves that event prediction methods which account for relational features between actors of social networks perform much better than conventional mainstream approaches like vector regression, random walk, markovian logic networks, etc. that are widely used today.

Certification of Thesis

This Thesis is the work of **Leonard Tan** except where otherwise acknowledged, with the **majority** of the authorship of the papers presented as a **Thesis by Publication** undertaken by the Student. The work is original and has not previously been submitted for any other award, except where acknowledged.

Principal Supervisor: **Associate Professor Ji Zhang**

Associate Supervisor: **Associate Professor Xiaohui Tao**

Student and Supervisors signatures of endorsement are held at the University.

Statement of Contribution

The following publications contribute to the thesis with the majority contribution for each paper from the candidate.

- **Article I:** Zhang, Ji, Leonard Tan, Xiaohui Tao, Xiaoyao Zheng, Yonglong Luo, and Jerry Chun-Wei Lin. "SLIND: identifying stable links in online social networks." In International Conference on Database Systems for Advanced Applications, pp. 813-816. Springer, Cham, 2018.

- **Article II:** Zhang, Ji, Leonard Tan, and Xiaohui Tao. "On relational learning and discovery in social networks: a survey." International Journal of Machine Learning and Cybernetics 10, no. 8 (2019): 2085-2102.

- **Article III:** Zhang, Ji, Xiaohui Tao, Leonard Tan, Jerry Chun-Wei Lin, Hongzhou Li, and Liang Chang. "On Link Stability Detection for Online Social Networks." In International Conference on Database and Expert Systems Applications, pp. 320-335. Springer, Cham, 2018.

- **Article IV:** Zhang, Ji, Leonard Tan, Xiaohui Tao, Thuan Pham, Xiaodong Zhu, Hongzhou Li, and Liang Chang. "Detecting Relational States in Online Social Networks." In 2018 5th International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC), pp. 38-43. IEEE, 2018.

- **Article V:** Ji Zhang, Leonard Tan and Xiaohui Tao, “Profiling Relational Turbulence of Students Using Adversarial Learning”, accepted by the Workshop on AI-based Multimodal Analytics for Educational Contexts (https://www.aima4edu.com/), joint iwth IJCAI 2019, Macao

- **Article VI:** Ji Zhang, Leonard Tan and Xiaohui Tao, “Learning Relational Fractals For Deep Knowledge Graph Embedding In Online Social Networks“, (**Best Runner-up Paper**

Award) the 20th International Conference on Web Information Systems Engineering (WISE 2019), Hong Kong, China, 26-30 Nov 2019

- **Article VII:** Ji Zhang, Leonard Tan, and Xiaohui Tao, “SLIND+: Stable LINk Detection”, accepted by the demo session of the 20th International Conference on Web Information Systems Engineering (WISE 2019), Hong Kong, China, 26-30 Nov 2019

- **Article VIII:** Zhang, Ji; Tan, Leonard, Tao, xiaohui, Thuan Pham "Recognizing Relational Intelligence in Online Social Networks" Journal of Computer Science Review (Accepted on December 12, 2019).

- **Article IX:** Zhang, Ji; Tan, Leonard, Tao, xiaohui, Lin, Chun-Wei; Zhu, Youwen Zhu "Event Prediction Using Fractal Neural Networks" Submitted to IEEE Transactions on Big Data (TBD-2019-03-0039) on March 24, 2019.

- **Article X:** Tan, Leonard, Hang, Kei Ho; Zhang, Ji; Tao, xiaohui "Predicting Events in Online Social Networks" Submitted to The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020) on December 25, 2019.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my research supervisors, Dr. Ji Zhang, Associate Professor and Dr. Xiaohui Tao, Associate Professor, School of Sciences, University Of Southern Queensland, Toowoomba, for this partnership and in providing invaluable guidance throughout this research.

I am extremely thankful to my parents for their sacrifices and eternally grateful to my significant other - my only true friend, who has relentlessly imbued me with the mantle to conquer setbacks and hard knocks encountered during this arduous journey. My Special thanks goes to my friend and brother in law Prof. Fred Mucholdyne for his support towards this thesis's success.

I would like to thank my friends and research colleagues, Dr. Hang Kei Ho, Prof. M.V.Seet, Prof. H.Soon, Prof. S.Thambi, Prof. C.Thomas Alderly, Prof. S.H.Goh, Prof.(M.D.) Jason Chen and Prof. W.Kai-shen for their constant encouragement. I express my special thanks to Mr. Thuan Pham, my co-author in arms, for his genuine support throughout this research work. I wish also to extend my very special, momentous appreciation to Dr. Hang Kei Ho for his dedication, consistant motivational support and moral advise whenever the hours became darkest during this journey.

I thank both Zhejiang research labs and Zhejiang Ivy research labs, Shenzhen for their instrumental support to the successful completion of this work. Finally, my thanks goes out to all the people - especially my students who have inspired greatness in me, daily. This research has been supported by an Australian Government Research Training Program Scholarship.

Leonard W.L. Tan

TABLE OF CONTENTS

List Of Figures

List Of Tables

List Of Abbreviations

- ANN - Artificial Neural Network
- DNN - Deep Neural Network
- DCN - Deep Convolutional Network
- GNN - Graph Neural Network
- GCN - Graph Convolutional Network
- FNN - Fractal Neural Network
- GAN - Generative Adversarial Network
- RNN - Recursive Neural Network
- LSTM - Long-Short-Term-Memory
- RFT - Relational Flux Turbulence
- RTM - Relational Turbulence Model
- RTT - Relational Turbulence Theory
- SNN - Spiking Neural Network
- CRF - Conditional Random Field
- RBM - Restricted Boltzmann Machine
- HMC - Hamiltonian Monte Carlo
- MCMC - Markov Chain Monte Carlo
- HMM - Hidden Markov Model

To God, my parents and Leona for their unyielding commitment and support.

CHAPTER 1

INTRODUCTION

Event Prediction plays very important and significant roles in many turn-key applications of information-networks today. Event prediction is a complex topic which encompasses a mix of multiple disciplines across wide ranging applications [2]. Some of them include recommender systems, marketing and advertising, governance and rule, news and propaganda, etc. [3]. Some examples of emerging event prediction applications include preemptive disease and medical condition prevention, patient-drug matching pair diagnosis and administration, cyber security, data privacy and utility, etc.

The social pre-cursors of a large majority of real life events are often staged through popular online social media like Facebook, Twitter, Google, etc. These pre-cursors are often identified as activities through online social mediums as information transactions [4]. Although it may be intuitive to think of a similarity based approach on how an actor influences other members within a community through matching attributes, such an aggregation of affective sentiments are oftentimes a lot less direct [5].

Structural stability in social networks has always been a topic of contention in various applications of interest. These include but are not limited to link predictive approaches, community detection methods and logical random graph models [6]. The key elements of relational stability have always been referenced to attributes perceived to be contained within links established between key actor / nodes of a social community structure [7].

Identifying relational flux and turbulence plays a pivotal role in determining structural stability of an OSN because it is able to determine the temporal relational states of actors within a social community of a given topic context [8]. These states can either stabilize or de-stabilize the social com-

munity in question [9] and may lead to detrimental event occurances in the future [4]. Many recent studies performed in this area of interest still rely on "flat" uni-directional linked structures that lead to inaccuracies and inconsistencies (instability) in the detected / predicted social structures and / or sub-structures [10]. A key observation that can be made through literature revolving around relational intelligence of a social network is an over-reliance on similarity measures between node to node or link degree features [11].

This research tackles the problem of describing relational flux and turbulence of three well-established major social networks (Google, Twitter and Enron emails) using principles of the Relational Turbulence Model (RTM) [12]. In this detailed study, the approach firstly examines ground truths proposed by the Relational Turbulence Theory [9] and adapts it to uncover evolutionary social transaction behaviors for event prediction. It also further develops the novel Fractal Neural Network (FNN) learning architecture to scale towards predicting different events through a series of temporal relational transactions in a vast social environment constantly evolving with sentimental and affective disruptions with topic drifts [13], [14]. This improvement in performance and accuracy is demonstrated in the results and discussion chapter of this thesis.

1.1 Challenges

Identifying relational flux and turbulence in OSNs is challenging for two reasons: its generalized, temporal evolving nature and latent implicit state transitions [15]. It is not a trival task to represent evolving communication behavior patterns Ψ between actors Λ of an OSN $\grave{G}$, and much less at describing relational state transitions $\varkappa$ as a time evolving flux F_{ϵ} of social transactions. Many relational approaches used in this study today, lack depth and representative power [8]. The drawback of these techniques is that important correlational attributes shared between actors are ignored, resulting in shallow representations of relational states [8]. Methods based on feature similarities throughout studies in literature, have shown the lack

of representational efficacy to model real life social structures effectively [11], [16].

Although numerous approaches have been developed to address certain areas of effective event detection and prediction, their methods have been limited in applications of specific events [4]. Furthermore, techniques to date focuses on the use of batch learning methods which can only be used at static instances in time [17]. Such approaches are known to be unscalable to continuous (social Knowledge Graphs) data streams and changing environment contexts [14]. Generally speaking, there are several critical key questions in this field of study which remain unanswered. In an unstructured social network within an evolving construct of dynamic relationships [8], firstly, how is it possible to represent generalizations of evolutionary behavior within these social transactions accurately? Secondly, how can dynamic relational profiles which correlate to different social communication patterns be recognizable? Thirdly, how accurate are predicted future events with respect to the relational features precipitating their occurrences? Finally, how quantifiable are the dynamic errors arising from social disruptions and topic drifts (outliers) in the predictions?

1.2 Data Model

This study addresses these questions with the use of Fractal Neural Networks (FNNs) within the Relational Turbulence Model (RTM) framework, which encode ground truths of the RTT construct into the lowest principle decompositions of the model. FNNs leverage on the dynamic structure of fractals as the lowest principle decompositions of never ending patterns. They are driven by a recursive process, and are adaptable enough to describe highly dynamic system representations [18]. FNN is able to self-evolve from a meta-learning perspective - in response to random "anytime-sequenced" data streams of fluctuating information sophistication [19]. Next, this approach defines what relational turbulence is and explains the substantiating motivation.

Firstly, this approach characterizes Relational Turbulence by probabilistic

measures of Relational Intensity $P(\gamma_{rl})$, Relational Interference $P(\vartheta_{rl})$ and Relational Uncertainty $P(\varphi_{rl})$ [20]. RTM defines an artificial construct (as a black box model), which predicts communication behaviors during relationship transitions in an environment of constant social disruptions [15].

Then, the principle of Relational Turbulence Theory (RTT) [9] is used to establish a framework of theoretical processes linking evolving relational features learned over past event occurrences (causals). They relate to relational reciprocity bias, sentimental and affective communication patterns, state altering events and role-recognition behaviors that identify relational uncertainty and interdependence [21].

This model was chosen as the main approach because alternative data models compromises accuracy and performance of predictions for simplicity in representation. Examples include node-based measures like node feature similarity and text feature similarity, neighbor-based measures like Common Neighbors (CN), Jaccard's Coefficient (JC), Adamic Adar (AA), etc., path-based measures like the Katz constant, LP, RSS, etc., random walk-based measures like SimRank, PropFlow, Rooted Page Rank (RPR), etc. [22]. These representations capture relational structures from a time static perspective and are not adaptable to real-life dynamic evolutions of relational states [23].

Generally for event prediction, the first approach category is the markovian sequenced model (also known as association rule based prediction) [24]. In this category, future event occurances are predicted based on past event association patterns. While this approch is able to capture temporal features relative to key (anchor) events, it assumes that events are correlated to each other in a fixed sequence. The second approach category is the stochastic word distribution model (also known as narrative generation) [25]. In this category, future event occurances are predicted based on the topic-context word distributions surrounding an actor in question. For example, when the name "Donald Trump" and the topic-context "President of the United States" is mentioned, there will be major events which are stochastically related (e.g. trade wars, tax tarrifs, mexico border, etc.). While this approach is able to draw a coreference resolution between word-topic to events, it overlooks the temporal aspects of such occurances [25].

In this work, firstly, focus is given to identifying stable links from temprorally changing relational features in links of a chosen Online Social Network (OSN). Then, attention is directed towards discovering relational intelligence through identifying relational flux and turbulence profiles on three major social platforms: Twitter, Google and Enron email datasets. Next, generalizations of event occurrences on three major social streaming platforms: Twitter, Google Feed and Live Journal are further developed.

1.3 Technical Model

This study firstly develops a stochastic model to detect stable links within a chosen Online Social Network (OSN). The model developed is known as the Multivariate Vector Auto-Regression (MVVA) and extracts relational features of links into a single regression model. The key objective of this model is to identify stable links from the temporality of these changing features. A significant improvement of the MVVA to other mainstream models is the efficient handling of dynamic link features in the prediction process. The Hamiltonian Monte Carlo (HMC) extension is further developed to this model to improve scalability to large datasets.

Secondly, this study introduces a new model - the Relational-Flux-Turbulence (RFT) that effectively represents the dynamism of popular key relational dimensions uncovered from previous approaches and techniques conducted on online social structures. This is done from the perspective of a time evolving flow of relational attributes (time-realistic relationships) between node entities of a network in question with the constant inception of social shocks. The model builds a multi-stage deep neural network from a stack of fractals with hybrid architectures of Restricted Boltzmann Machines (RBMs) and Recursive Neural Nets (RNNs).

These structures are self-evolving from a meta-learning perspective. The neural network accepts as inputs, key relational feature states f_i between actors a_j and global events E_ϵ from past and present social transactions to determine the likelihood of relational turbulence τ_{ij} within an identified social

flux F_ϵ. Relational turbulence may correspond to various disruptions in social communication of different environments and contexts [9]. For example, in the discussion of world events like trade wars, passive sentiments passed through public posts and comments are indicative of hostility and potential conflict which may lead to a breakdown of linked integrity between actors in many aspects like trust, influence, status, etc.

In addition, as a major contribution, this thesis also shows that time evolution relational flows improves efficacy of existing approaches through studies and comparisons of experimental results conducted on real life social networks. This study develops a novel architecture from Relational Turbulence Theory and Models (RTT and RTM) to identify social disruptions by estimating relational turbulence profiles, within a given social context describing the state of flux. Then, this approach evaluates the methods on Twitter, Google and Enron email datasets and demonstrate that they outperform similarity based feature and shallow uni-directional flat structural approaches in detecting social flux and turbulence.

Next in this thesis, the Relational Flux Turbulence model (RFT) for event prediction is presented, which is developed from the principles of self evolving fractals and artificial neural networks in a real-time machine learning model for active data streams [26], [18].

Its objective function describes the turbulence profiles of social graph constructs and their resulting communication behavioral patterns across apriori relational state altering events - to predict likelihood occurrences of tracked topics as events of interest.

This method accepts as inputs the concurrent key relational feature states f_i between actors E_ϵ from past and present social transactions to predict the likelihood of an event occurance E_φ in an evolving state of relational turbulence τ_{ij} from an identified social flux F_ϵ within a continuous stream of social transactions [27], [28]. As a major contribution of this thesis in the later chapters, further evaluations of the methods on Twitter, Google-feed and LiveJournal datasets are detailed and demonstrated to outperform HPM and other state-of-the-art models [1].

1.4 Research Methodology

The research problem can be formulated as follows: How can we predict general world-wide events of interest accurately and reliably from current geo-political states of evolution? To properly address the problem of event generalization, we approach this study from a perspective of computationally understanding and representing relational attributes and their behavior across several interesting geographically world-wide events and their occurances. The key objective of this study is to predict heterogeneous events without compromise to performance of the developed model. The choice was made to study relational models instead of node / entity models because importantly, latent information about relational states and communication patterns during events are important pre-cursors to the occurances of these events in question. This thesis differentiates itself from other mainstream research because its focus is on relational and graph structural approaches and not as such a node-entity approach - as is commonly used in current literature concepts. Additionally, in order to adequately describe and represent knowledge structures of chaos, this study develops deep learning models, designs, techniques, methodologies and approaches which are built on the mathematical concepts and foundations of fractals. This research provides a wide range of breath and depth discovery into relational turbulence and event prediction.

Firstly, the opening chapter performs a detailed study into the techniques, approaches and methods surrounding relational pattern recognition of social network models and graphs. The extensive overview of heterogeneous information architectures and machine learning techniques which are used to uncover latent knowledge and features provides a firm and solid foundation for the development of the approach.

Secondly, the study progresses towards describing and representation of relational states of heterogeneous Online Social Networks. As the first milestone, stochastic methods were developed and evolved from the task of detecting relational stability in OSNs using the MVVA technique. The Monte Carlo extension provided further confirmation that this approach can be adopted on larger scale social structures at the cost of computational power.

As the second milestone of this research, an RFT model was developed from the principles of relational turbulence and the Fractal Neural Network. This approach leverages on relational state transition behaviors observed by the RTM principle during eventful occurances and uses the FNN to represent comlex and sophisticated relational evolutions between actors of a social network. It shows from results which follow, that the RFT model is able to scale towards highly complex relational representations which closely resembles real life social structural interactions. Furthermore, the highly adaptive FNN is also able to present these representations efficiently in real time streaming data like twitter. In this method, this technique offers an alternative to current Active Online Learning methodologies covered in the presenting chapters of this thesis.

As the third and final milestone of this research, this thesis develops a generalized event prediction model from the experimented RFT architectures. In the model design, topic modeling and wavelet transformation were used to detect events from a continuous stream of social transactions. Queries were made in two successive social streams where relational states and their behavior of specific eventful topics were profiled for event prediction tasks. In these eventful social transactions, communication patterns corresponding to their social relational states were learnt by the model. As an extension to the design of RFT, this study also includes an adversarial model to effectively predict events accurately and reliably. In the following results, RFT is shown to fare much better than current benchmarks on the scale of predicting events in the absence of a markovian framework. As a result, these developments offer a wider, more generic approach to identify, detect and predict the occurance of events from OSNs.

1.5 Contributions

The approach examines the dynamic structure of such an shallow ANN known as fractals. Fractals are the lowest principle decompositions of never ending patterns. They maintain a key property of self-similarity across different varying scales [26]. A careful distinction here requires constant discriminations between similar and identical schemes. Fractals generally maintain

structural affinity [29] and can grow to become complex enough to represent high levels of sophistication that are yet trivially efficient enough to re-create by repeating similar simple shallow architectures in a loop - ad infinitum.

Driven by a recursive process, fractals are adaptable enough to describe highly dynamic system representations [18]. In the chapters that follow, descriptions on the methods and experiments performed on Twitter, Google and Enron email datasets at different instances are given and shown that structural fractals behave like cognitive super primers that can be used to decode representational information sophistication through a generative feedback loop. The main scientific contributions of this work are presented as follows:

1. A method to adaptively learn from real-time online streaming data to identify turbulent relationships within a given OSN was developed for real-time data processing applications;

2. An innovative RFT model was developed to capture key relational features which were used to detect and profile social communication patterns of eventful states within a given OSN for Graph Evolutionary Networks;

3. The event prediction approach adaptively learns from a Fractal Neural Network (FNN) which builds on key relational fractal structures discovered in a given Online Social Network (OSN) from tracked topics to represent complex relational states of Online Social Networks.

4. An innovative adversarial FNN framework architecture is then used to accurately and efficiently predict future occurrences of similar events for general event prediction applications.

5. The approach shows that the RFT model improves the efficacy of existing approaches towards event prediction through studies and comparisons of experimental results conducted on real-life social networks.

6. The experimental design and detailed results show that RFT is able to offer a good modeling of relational ground truths, while FNN is able to efficiently and accurately represent evolving relational turbulence and flux profiles within a given OSN.

The remaining part of the thesis is organized as follows: Chapter II presents detailed reviews of related works drawn from social theories and relational structures. Chapter III elaborates on key concepts, theories and preliminaries of link stability prediction in OSN. Chapter IV extends the methods and models which have been developed for identifying relational flux and turbulence in OSNs. Chapter V highlights the novel implementation architecture from this research to address the problem of event prediction. Chapter VI presents the concluding remarks and discussion of this thesis that leads to a finale and potential future directions in section VII.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Some of the biggest problems with technology mediums today, are with stable, effective, communication and control of highly complex yet dynamic distributions of real time / life information [30]. Such naturally occuring behaviors result from constantly evolving relationships [31]. They maintain key characteristics over time, that generalize into a breakdown of hierarchical organization and centeralized control. These topologies however, are not completely regular or random. Many information networks of real life systems (e.g. biological, technological, social networks, etc.) lie somewhere in between the twin extremities of structural regularity and randomness [32]. Challenging research problems in this field of study, addresses the universal need to discover, recognize and interpret as much latent information as possible from relational patterns [33]. The objectives, are to more reliably and efficiently fulfill big data tasks through machine recognition. Examples of such tasks include data mining [34], online recommendations [35], event prediction [36], privacy and security [37], call routing and traffic control [38], bandwidth allocation [39], logistics and planning [40], protein synthesis, cellular regeneration, epidemic spread and control, stem cell culture, treatment of affective and mental disorders, Arrhythmia, Cardiomyopathy, Cystic Fibrosis and Lymphoma, Immunogenetics and Disease, etc.

Online media by far, offers the largest concentration of social information. One of the most pressing concerns over its future directions of use from current observations of detriments is the dramatic rise in cybercrime [41]. The most severe socio-economic impact and consequences revolve around identified cybercrimes such as: online fraud, cyber stalking, hacking, phishing,

cross-site scripting, vishing and botnets [42]. Across prominent countries like the United Kingdom, Germany, Netherlands, Sweden, Switzerland, Australia, United States, Canada and Hong Kong, cyber fraud incidents have been reported to have risen by almost 100% of its national crime statistic since 2015 [41].

This literature review was conducted with the following objectives in mind:

1. Firstly, to capture general trending approaches to computer recognition of key relational patterns and complexities revolving around information networks.
2. Secondly, to uncover relational intelligence behind evolving structural topologies giving special focus to Online Social Networks (OSNs).
3. Lastly, to provide insights and discussions into the emerging trends of Online Artificial Intelligence (OAI) as a future research direction for pattern recognition in information networks.

The remaining part of this chapter is organized as follows. Section 2 presents the detailed scientific contributions of this study. Section 3 covers fundamental theories and overviews of relational pattern recognition processing models. Section 4 discusses and reviews relevent principles and problems of data acquisition and pre-processing tasks. Section 5 elaborates on methods of identification from acquired and pre-processed data. Following section 5, section 6 provides a thorough analysis on high level concepts of classification and prediction from information extraction tasks. A significant highlight to discussions in all domains include critical analyses and comparisons on recent research. In addition, the importance of integrity in structural patterns to their corresponding models are also featured. Section 7 elaborates on some more advanced topics of OAI, its role in the machine recognition framework and sets the pace for future research within the field of OSNs. Finally, Section 8 concludes this chapter with a thorough discussion on trending directions.

2.2 Scientific Contributions

The research challenges of this chapter can be formulated as follows: given a richly dynamic, set of social transactions, can a system be designed to continuously and adaptively recognize ubiquitous relational intelligence from the social patterns which have evolved? Additionally, how can the system interpret contributions of these socially encoded features toward certain end applications - like the prediction of human events [43], [44], [45], continuous complex emotion recognition [46], [47], [48], etc.? There have been several developments in this field of study like those surveyed in [49], [50], [51], [23], [52], [53] and [6]. However, their methods of analysis are focused directly on a specific area of interest. For example, [23] and [52] explores the problem of link prediction directly in relation to their underlying social patterns. These surveys identify recent advancements in techniques that tackle increasing relational complexity. In their works, they highlight how simple use cases of predicting links in "flat" homogeneous social structures like Node, Neighbor, Path, Random walk and Social Theory Based approaches would fail in a more realistic heterogeneous environment. Although prediction methods used in their approach infringe onto relational patterns of information networks, it often overlooks the intelligence behind adaptively recognizing these relations and ties as a critical part in their analysis. For example, the Preferential Attachment metric [11], sampled by the authors of both papers neglects the reliability of higher degree order nodes when predicting links. In the same vein, Resource Allocation (RA) [51], Adamic Adar (AA) [54], Jaccard Coefficient (JC) [55], metrics etc. all work on similar underlying assumptions that uniform relational patterns pre-exist in information networks. Even for time aware applications like [56], [57], [58], [59] and [60], relational patterns of these information networks in their study have not been fully qualified nor quantified (recognized) before suitable predictions are made.

However, in contrast, surveys conducted by [50], [53] and [6] provide recent research work on relational patterns of highly complex heterogeneous networks. Their review uncovers rich latent semantics and inferences of meta-objects and links within complex networks. Their objective is to summarize key methodologies for effectively recognizing and mining useful knowledge from information networks. Some applications covered in their study in-

clude techniques like link prediction, community detection, recommendation models, influence, trust propagation and rumor spreading. In similarity, all three papers address an important relational structure of Online Social Information Networks: heterogeneity and its correlation to both complexity and large data volumes. In difference however, [50] and [53] focus on the heterogeneous application tasks like Classification, Clustering, Similarity Measures, Prediction, Ranking, Recommendation, Fusion and Enrichment, etc., while [6] covers more important pattern recognition aspects from a stochastic perspective. Uncertainty in their study, is modeled probabilistically as a joint event of relational pattern reference and existence [6]. Statistically, it is used as a powerful method of approach to predict unknown distributions in a future time frame. However, a major flaw of stochastic models is that they require special attention to detail over how thresholds should be parameterized so that convergence is guaranteed. This flaw is inherent because of the lack of fundamental principles governing ground truths of most unstructured information networks - especially OSNs.

This thesis chapter provides a comprehensive framework for the recognition of relational patterns in OSNs which is missing in previous most recent reviews. To the best knowledge of current literature, there has not been a survey conducted on general recognition tasks for OSNs - which this chapter objectively illustrates, through the use of well known models at each stage in a logical process flow. Furthermore, this chapter also defines the novel concept of relational intelligence as evolving link patterns observed within a social structure and build a model for recognition tasks which have often been overlooked in recent literature. The major contributions of this review are as follows:

1. Firstly, a development of the general framework model is provided for the affective and sentimental recognition of relational patterns from OSNs over all surveyed methods covered in this study;
2. The second major contribution of this review chapter uncovers key underlying intelligence of relational patterns in information networks. The developed approaches from this study can then used to tackle real-life issues related to privacy and security of OSNs (e.g. Cyber Fraud);
3. Lastly, a third major contribution of this survey chapter identifies main problem areas of current approaches in recognition based tasks like pre-

diction, detection, recommendation, ranking, etc. In addition, future trends and directions of research in this field of study are highlighted to tackle the needs and problems faced in OSNs and Social Internetworking Scenarios (SIS).

2.3 General Overview

A process of recognizing relational intelligence from patterns can be broken down into two broad phases. The first phase is often concerned with acquiring, handling and managing data volumes efficiently and effectively [61]. This enables most baseline models and algorithms developed in this field of study to adequately package interesting information for retrieval (IR). IR in turn is used to satisfy the utility requirements of end applications [62]. The second phase is concerned with the higher level interpretations of information transitioned from IR processes [63]. Its core purposes serve to empower more advanced tasks like, prediction, recommendation, anonymization, etc. The major steps involved through the whole recognition process include the following:

1. Data acquisition and Pre-processing [64], [13]
2. Identification and Extraction [65]
3. Detection and Labelling [10]
4. Classification [66]
5. Learning and OAI [67]
6. Performance Evaluation [68]

A detailed diagram of the general recognition framework used in parts of the entire study of OSNs is given in Figure 2.01.

The various modules illustrated in Figure 1 will be elaborated on in the sections that follow this discussion. Pattern recognition processes can be viewed as sequential signal convolutions of structured mechanisms that perform sensing, processing, learning and behavior influencing functions [69]. Pattern recognition itself is defined as the cognitive phenomenon of identifying physical changes within an environmental world of existence and associating with the adequate reactions correspondingly [70]. A pattern is defined as a set of structured changes resulting from a system of continuous exter-

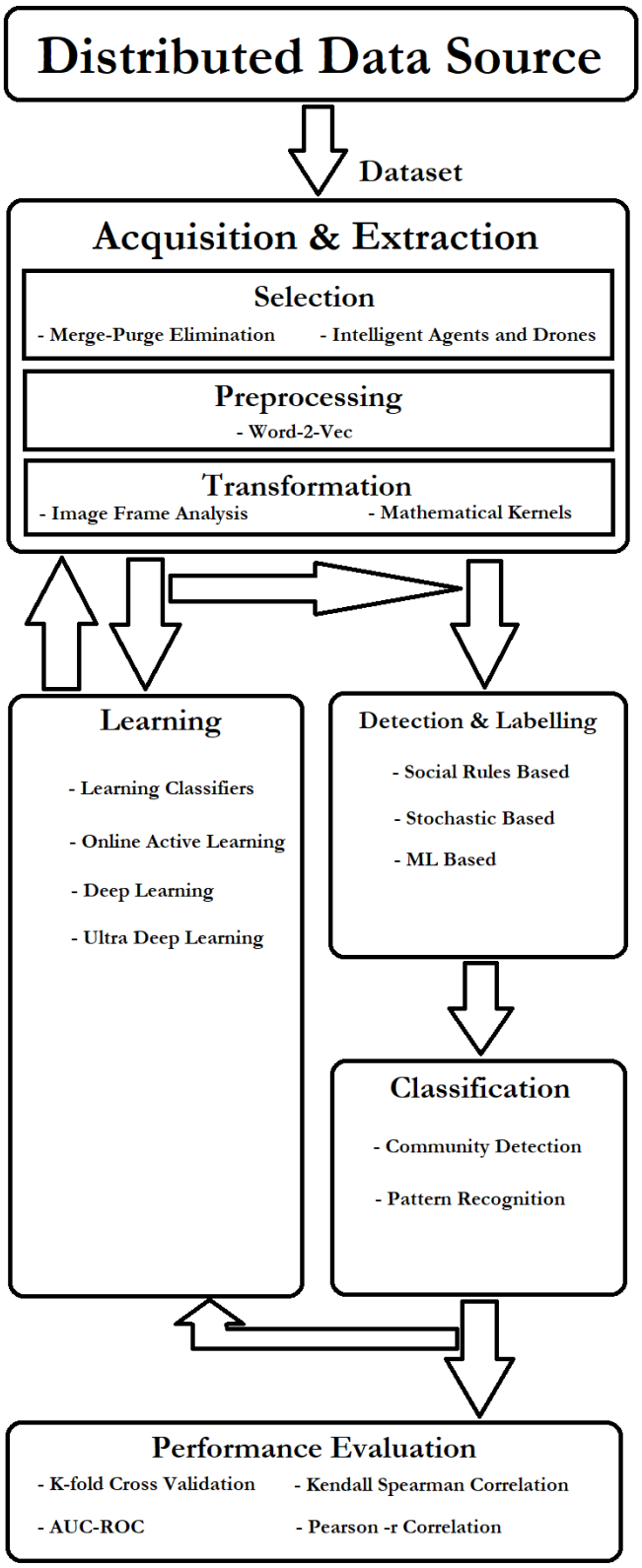


Figure 2.01. Recognition framework model for relational intelligence of patterns in OSNs

nal interference to its localized steady state signal [71]. The principles of modelling pattern recognition as a structured process contains four major categories of key considerations. They are widely defined as:

1. Bayesian Decision Theory [72]
2. Statistical Classifiers [73]
3. Dimensionality Reduction [74]
4. Clustering [75]

This section briefly introduces some of these considerations as guiding principles to our general pattern recognition framework model built for OSNs.

2.3.1 Bayesian Decision Rule

The Bayesian is concerned with reasoning of very complex tasks which involve high level inferences and planning [76]. Bayesian decisions form a subset of inference mechanisms which are more powerful and flexible than other decision rules within its own category. However, a main drawback of this model is utility at the prime cost of speed [77]. A bayesian decision describes how likely (or unlikely) an event would occur based on the conditions of prior related knowledge [78]. Mathematically, it is described as:

$$P(w_i|x) = \frac{P(x|w_i)P(w_i)}{P(x)} \tag{2.1}$$

Where w_i is an event outcome to be probabilistically decided and x are observations of conditions related to past occurances of such events w_1, w_2, w_3, etc. Intuitively, a good decision would be to choose an outcome w_k with the largest probability of occurance. This can in turn be described as:

$$k = sup_{i \geq 0} P(w_i|x) \tag{2.2}$$

The main benefit of the bayesian maximal probabilistic approach is its average minimum decision error yields.

Minimum error rate decisions extend Bayesian rules by making choices that maximize posterior probability functions. This thereby means that errors in decision making would be probabilistically minimized when choosing an outcome at which there are highest densities of "ground truths" in its

occurance [79]. The intuition behind this is that for a given set of event categories to be predicted $\nu = w_1...w_s$ over a possible set of predictions $\gamma = \delta_1...\delta k$, it objectifies an expected loss tolerance from the prediction process $l(\delta_i|w_j)$ and look to past observations x as evidence to formalize ground truths by event occurance densities [78]. Then, a rule $\delta(x)$ is sought out - which minimizes the overall risk function of the decision process:

$$R = \int_{-\infty}^{\infty} R(\delta(x)|x)P(x)dx \tag{2.3}$$

This can then be reduced to:

$$\sum_{j \neq i} P(w_j|x) = 1 - P(w_i|x) \tag{2.4}$$

Which essentially states that in order to minimize decision error probabilities, the largest posterior categories of outcomes must be chosen because they contain the densest probabilities of "ground truths".

2.3.2 Statistical Classifiers

Statistical classifiers form a large body of stochastic approaches that seek to accurately decide on the outcome of an event when there is very little evidence of "truth" distribution patterns from past observables [80]. In other words, for most practical scenarios, $P(x|w_i)$ is unknown. Due to these unknowns which are essential to a decision making process for recognition, the wide use of statistical classifiers employ a rich variety of kernels [81]. Kernels are masks that transform linear objective functions into desired non-linear representations which effectively perform similar tasks like classification, prediction, decision support, etc. [82] An example of such a classifier is the Quadratic or Gaussian classifier [83] which assumes that the likelihood function $P(x|w_i)$ follows a Gaussian distribution:

$$P(x|w_i) = \frac{e^{-1/2(x-\mu_i)^T \sum_i^{-1}(x-\mu_i)}}{(2\pi)^{n/2}|\sum_i|^{1/2}} \tag{2.5}$$

Where μ_i is the distribution mean and n is the observation sample size. Other large collections of statistical classifiers are logic-based [84]. One example of

such a classifier is the K-nearest neighbor (KNN) [80]. KNN is an intuitive detection and labelling mechanism of many OSN recognition models that is based on neighbor similarity displacements. It aims to find a "closeness" euclidean measure between a node of interest and its proximity of neighbors [85]. It is simple to implement and analyze [86]. However, KNN does not scale well to increasing complexity of feature dimensions [86], [85], [87].

2.3.3 Dimensionality Reduction

Dimensionality reduction is a necessary process in most big data recognition frameworks that tackles the problem of learning and trainability of the model in the design [88]. Essentially, dimensionality reduction is often seen as a drawback to most system architectures because it eliminates data which may be oftentimes regarded as 'important' to its application's utility. However, on the other hand, dimensionality reduction is necessary for control and manageability of intermediate processes like identification, detection, learning, etc. within the machine recognition framework [88], [74]. Furthermore, it also avoids the risk of overfitting data into a particular model. As a core pre-processing step (which will be elaborated in the next section), when dimensionality reduction is used adequately on relational pattern features that contribute little to the mapping of the underlying social structure, excellent accuracies resulting from recognition tasks more than compensate of the loss of unrelated feature data [13].

2.3.4 Clustering

Clustering is a broad area in the study of the classification process that is essential to most decision making techniques [89]. This includes high level interpretation capabilities of recognition tasks [75]. There are many approaches to classifying data according to its concept of use - which will be explored in later sections of this study. However, they can be widely categorized as supervised and unsupervised learning methods [90]. The general objective function of clustering is to segment data into labeled classes which are then used for the higher level inference mechanisms within the recognition processes. The goal of clustering algorithms is to build a model that captures

the structure of the feature data in question [91]. Some examples of these would include the Gaussian Mixture Models (GMMs) - which is a stochastic based approach [92]. This approach segments data into their respective class labels by modelling probability density functions with mixtures of parametric densities. Generally, they are described as:

$$P(x|\theta) = \sum_{z=1}^{Z} P(x|\theta_z)P(\theta_z) \tag{2.6}$$

Where $x \in X$ is the bag of training samples and θ is the parameter of estimation. Another example in this area would be the K-Means clustering - which is an extension of the key concepts used in the KNN classifier [91]. It aims to find a "compactness" within a group of features through proximity measures in order to segment them into group membership labels. This metric is expressed as:

$$inf_{w_z} = \sum_{z=1}^{Z} \sum_{x \in w_z} ||x - \mu_z||^2 \tag{2.7}$$

for

$$\mu_z = \frac{1}{n_z} \sum_{x \in w_z} x \tag{2.8}$$

Where μ_z is the membership classifier label (usually the mean point) over n number of neighbors and w_z is membership category set.

To summerize, computer pattern recognition is a machine cognitive process which associates realistic characteristics to observed patterns as sets of identifiable properties in a system. We define that every relational concept is a set of properties, then a set of patterns.

2.4 Data Acquisition From OSNs

Data inter-relationships play very important roles in determining how knowledge is correlated to each other in a highly dense manifold. Such correlations between entities determine patterns of behavior over time which is then oftentimes used to describe the evolutionary nature of these relational structures

[93]. The importance of such correlations span wide areas of interest which include, but are not limited to: social ties and structures, biological networks, computer and communication systems, transport, air routes, etc. [94]

Previous works however, have all assumed the use of certain ground truths that correlate to prior observed structures and their patterns. Although generally accepted by the community to be an adequate baseline of measure, several key questions concerning the acquisition and pre-processing of such derived structures remain unanswered. Some of the attempts aimed at addressing a few of these observed structural patterns are through the use of complex schemas (e.g. Bespoke-Star, Multi-Relation, Bipartite, Edge-Node (multi-hub), etc.) [50]. A key drawback of such schema classifications is that they are highly biased towards a small locality of the network [6]. A prime example is that in the analysis of such schemas, weak links are oftentimes ignored and overlooked as trival [95], [96]. Although analytically, they do not play a direct role towards attributing the importance scores of strongly intra-connecting links of nodes (defined commonly in literature as communities) within a very limited window of study, it is argued that through time, weak ties preceeds strong relations in order of ranking where noval information flows are studied [95]; hence thereby, indirectly changing the dynamics over how social structures are formed and re-formed.

Data mining techniques and methodologies span a wide array of approaches over vast online media sensory modalities [94]. They address both knowledge acquisition and pre-processing from unstructured information sets in OSNs. The need for improving data mining approaches is driven by the exponential growth in innovation of new computational theories and tools [97]. Some of the more low level applications in data mining are given below:

1. Data mining methods for pattern discovery and extraction (from low level - high volume datasets) to assist in the knowledge discovery process.
2. Automation of data acquisition and intelligent pre-processing through use of mathematical relation and methodologies / theories.
3. To effectively manage and warehouse increasing database sizes to combat data overload.
4. Homogenizing fragmanted data sets accumulated in cloud space for

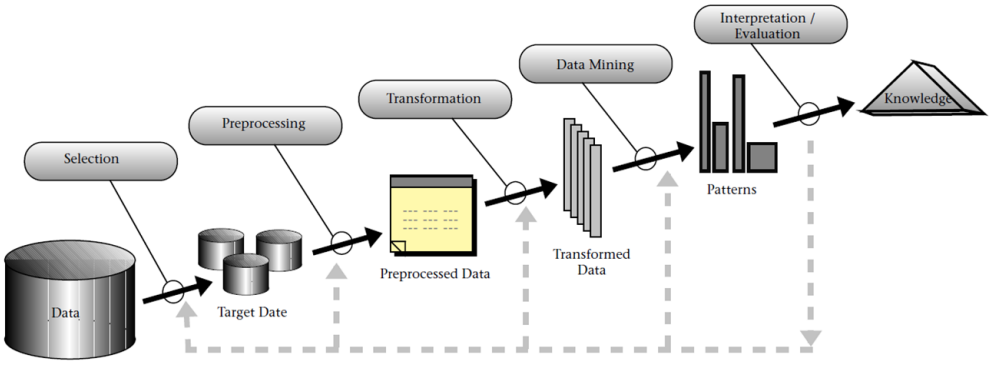


Figure 2.02. The KDD process [49]. The figure shows the stages of which data is successively passed through from selection / sampling to preprocessing, transformation, data mining and finally interpretation.

Type	**Modality**	**Technique / Method**
Data Pre-processing	Text	Word-2-Vec
Data Transformation	Camera	Frame-based image analysis
Data Selection	Online Media	Merge-Purge elimination
Data Selection	Online Media	Intelligent agents and drones (web crawlers)
Data Transformation	Online Media	Multidimensional data analysis and transformation

Table 2.01. Table of data mining techniques and methodologies

intelligent analysis.

A diagram illustrating the stages in the knowledge discovery and data mining (KDD) process is given in Figure 2.02.

The data acquisition process involves three other core sub processes at the lower levels to structure data according to its intended discovery objectives. Selection, preprocessing and transformation re-arranges data into discrete formats that can be easily sampled and managed [98]. A summary of data mining techniques and methodologies is given in the Table 2.01. Once data has been acquired and pre-processed, it is then passed onto the higher layer interpretation stages for identification and extraction.

2.5 Pattern Identification and Extraction

Information identification and extraction techniques fulfils mid-level objectives of finding and understanding social relational patterns in relevent parts of a given graph [99]. The goals of pattern identification and extraction are to produce a structured semantic representation of all information of interest in a precise form so that further inferences and analysis can be made about the obeserved relational intelligence from the topology [98]. Generally, relational pattern identification and extraction of OSNs can be categorized into three broad areas of study:

1. Social Rules Based Approach
2. Stochastic Approach
3. Machine Learning Approach

This section provides a detailed review to the various approaches used in recent literature for information identification and extraction of various social structures.

2.5.1 Social Rules Based Approach

A rules based approach describe social properties at the lowest representational levels based on theoretical formulations. Information which conforms to these rules are extracted through pattern matching techniques. Such rules are suitable for extracting interesting features from predictable relational behavior [34]. Recently, numerous studies have turned to social theories in their attempts to justify models developed for representations of high dimensional relationship patterns from theoretical perspectives of "ground truths" [61], [100], [101]. Some of these theories include: community modularity, triadic closure, strong and weak ties, homophily, structural balance and status. These theories seek to explain the behavior of social interaction between nodes from a psychological perspective and thus provide further insights into how relational structures are formed [102], [103], [104].

The social theory behind relational patterns formed within communities define a highly intuitive construct of node /actor behaviors [105]. These behaviors are a densely directed set of high frequency interactions centered around shared expectations, attributes, characteristics and features [31]. More

importantly, as a relational model, relational patterns within a community are often modeled as ties which strongly bind nodes together within a group [106]. These relational links provide for reliable channels of information exchange [107]. The strength of such links measured as:

$$\varphi_{k_{ij}} = \alpha \frac{[\sum_{i,j=0}^{n} k_i k_j] \otimes [\sum_{i,j=0}^{n} (A_i \cap A_j)]}{2m \sum_{i,j=0}^{n} (A_i \cup A_j)} \tag{2.9}$$

Where $\varphi_{k_{ij}}$ is the measured strength of a link between nodes i and j. k_i and k_j are adjacent dyadic degrees of node i and j respectively while $A_i \cap A_j$ denotes the common node attributes identified between 1st degree neighboring nodes i and j and $A_i \cup A_j$ denotes the 1st degree community sum attributes of both nodes i and j. $2m$ represents the likelihood estimation of the total number of 1st degree dyadic pathways to immediate neighbor nodes of i and j respectively.

Triadic closure theory defines a transitive property of among three nodes to form a closed triadc loop of social ties [108]. It is a belief inference that between three nodes, A, B and C; if a strong tie exists between nodes A and B and a similar relationship exists between nodes B and C, then there must be - by association, that a link also exists between nodes A and C. A drawback of this social theory is that it ignores the attributes of link features needed to form a high dimensional relationship. It is therefore, not practical enough to measure up against the ground truths of highly complex and data volumic (large) networks [109], [110], [20]. In most use cases, triadic closure acts as a model simplification of high level networks that is often used to predict missing and spurious links [111]. It achieves this by defining a clustering coefficient among tuples of nodes as a likelihood measure of the probability that three nodes will form a triadic relationship. This is given mathematically as:

$$Clust(G) = \frac{1}{N_j} \sum_{i \in v_i, d_i \geq j}^{n} C_i \tag{2.10}$$

Where N_j is the number of nodes with degree at least above a predefined threshold j. C_i is the clustering coefficient of node i and is defined as:

$$C_i = \frac{\delta(i)}{\tau(i)} \tag{2.11}$$

$\forall\ \delta(i)$ is the total number of tessellations formed over the total number of node i tuple $\tau(i)$.

Homophily measures the tendency of nodes to form ties with each other given similarity metrics which define the propensity of bond associativity [112]. Such a measure often reduces to a cosine similarity score between a vector of features obtained between node attributes. It is generally expressed as:

$$S(i,j) = \frac{|\vartheta(i) \cap \vartheta(j)|}{\sqrt{|\vartheta(i)|.|\vartheta(j)|}} \tag{2.12}$$

Where $S(i,j)$ is the cosine similarity measure between nodes i and j; $\vartheta(i)$ is the membership attribute of node i, and $\vartheta(j)$ is the membership attribute of node j.

Structural balance and status theories introduces signs to links within network tessellations that describe social dominance of actors / nodes in a triadic relationship [113]. From a signed directed graph, it is socially inferred that a positive link from node A to B indicates that B is of a higher social status to A. Similarly, a negative link from B to C infers that B is of a higher social status to C. This triadic relationship also reflects on the existence of social balance if a positive sentiment valence reciprocity exists between two pairs of nodes [114]. For two nodes i and j, sentiment valence measures the polarity of sentiments (negative or positive) shared between them. While reciprocity measures this net exchange - so that C_{ij} defines the net one way exchange of sentiments (both positive and negative) from i to j. Therefore, it implies that perfect reciprocity exists when $C_{ij} \equiv C_{ji}$. This means that if the dyadic relationship polarities between nodes A and B are perfectly reciprocal, then the relationship between A and B is said to be socially balanced. Such a structure is given in Figure 2.03.

2.5.2 Stochastic Approach

Stochastic approaches provide flexible and versertile capabilities to model highly complex relational patterns across multiple planes of representation [95], [115], [96], [116]. Such relationships are also known as hyperlinks in a socio-graph. They provide the adequate representational power to uncover

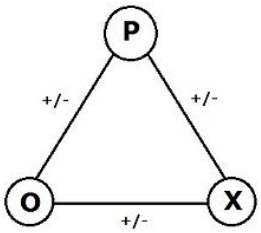


Figure 2.03. Basic Triadic Friendships. The diagram shows the effect of information transaction transitivity on status, balance and reciprocity.

latent intelligence within the complexity of social relationship patterns [117]. This provision improves both accuracy and fidelity of feature information extraction and identification processes in OSNs [118]. Prominant methods in this area include the wide use of the Bayesian model, Hidden Markov Model (HMM) and Decision Trees for probabilistically identifying and extracting relational information from patterns.

Heterogeneous structures evolve from hyperlinks that contain high dimensional features and traits which describe complicated relationships. Such relationships and their evolution affect the core mechanics of information transfer and flow in social networks [119], [50], [120]. More recent approaches towards addressing the representation of hyperlinks include segmenting these complex hierarchical structures into multiple relational planes of information exchange [121]. Individual layers contain a unique set of relational schemas between a subset of nodes within a topic context. An example of such a structure is given in Figure 2.04.

It is not a simple approach to treat each heterogeneous problem structurally as an independent set of linear equations to a solution [50]. Pre-existing conditions like hidden correlations between the same object groups (that in more recent study have been uncovered by Latent Dirichlet methods) have to be taken into account [122]. Latent Dirichlet (LD) methods seek to capture hidden dependencies among entities modeled with a plane notation [123]. The technical implementation of this method with respect to information networks is identical to probabilistic latent semantic analysis (pLSA) solutions [124]. The only difference is that LD assumes a sparse prior distribution whereas LSA establishes a dense apriori sampling distribution [124], [125]. As a baseline model, the LD is an extension of the pLSA, which

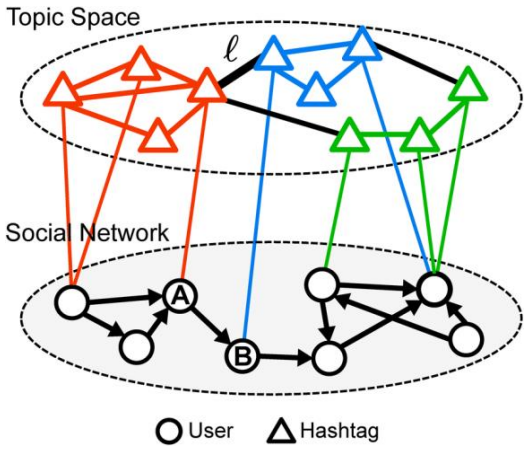


Figure 2.04. A Multi-Layered Social Network. This diagram shows different logical planes of representation for high dimensional features that map onto unique entities.

is defined as:

$$P(e,i) = \sum_{c} P(c)P(i|c)P(e|c) = P(d)\sum_{c} P(c|i)P(e|c) \tag{2.13}$$

It describes the co-occurrence of a node attribute i with a relational link e across the relational planes where it has been represented. The LD extends this basic representation through Bayesian inferences by assuming a generative process for each link-node occurrence pair [126]. It first chooses a discrete normal distribution followed by a Dirichlet distribution. Then it samples node-link pairs from the posterior multinomial distribution [124], [127]. The probability density function of the LD prior is given by:

$$P(\theta|\alpha) = \prod_{j=1}^{k} \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_j^{\alpha_j - 1} \tag{2.14}$$

Where the k-vector α contains components $\alpha_i > 0$, and the gamma function is given by $\Gamma(x)$. Given the parameters α and β (link to node attribute probability matrix), the joint distribution of the co-occurrence mixture θ, a set of N nodes z and a set of N relations e is given by:

$$P(\theta, z, w|\alpha, \beta) = P(\theta|\alpha) \prod_{n=1}^{N} P(z_n|\theta)P(w_n|z_n, \beta) \tag{2.15}$$

A significant research challenge of recent studies involves linking distributions

of sub-graphs with hypergraph structures [128]. This problem is compounded by further aligning them together with these co-occurances [129]. One reason for this complication is due to the lack of truth observations between the relational layers of the topology [128]. Another reason is because of incomplete / sparse data representations at the structural layer [128]. As a result, this prevents planar reconstruction of nodes and links to form an adequate relational sub-network for study.

Proximity is quantified in [51] as a measure of the closeness in similarity between node attributes. This enables a characterization of the likelihoods in estimating the prediction of links in online social networks. They propose two novel techniques to address the problems of efficiently and accurately estimating proximity in large dynamic social networks within the order of millions of nodes (a hypergraph). Their methods consider two variants of link predictors. The first are basic link predictors, which reduces the networks into a one-dimensional layer. While the second are composite link predictors, which establishes hyperplanes of links between several measured attributes of proximity. They use a powerful summary technique called sketch to reduce the dimensionality of heterogeneity by projections along random vectors. Their approach is closely related to the count-min sketch algorithm used in several other studies [130]. A common problem of the proximity inversion method that the authors employ in their paper is the imbalance in matrices between the proximity sketch M and the inverse matrix P defined as:

$$P = \triangle(1 - \beta M)^{-1} = \sum_{l=0}^{\infty} \beta^l M^l ... \text{ As seen in [58]} \tag{2.16}$$

The authors address this by developing two dimensionality reduction techniques to approximate elements of P based on static snapshots of M. They then incrementally update the proximity algorithm to include elements of P in an online setting as M continually evolves. However, as a data dimensionality technique, their sketch algorithm suffers from accuracy since it is sensitive to complexity. The inaccuracy of their proposed method grows exponentially, as the matrix rank of P increases. Hence, their link predictive approach degrades as proximity measures grows.

In evolving, dynamic networks, temporality plays a very important role in determining the type of relationships formed and re-formed over time [35], [131]. Structural reliability of the network over time can be stochastically estimated from simple contributing relational features - like link co-occurances across time frames. A Bayesian network can be described by a directed interconnection of states as "nodes". These nodes which have been identified and assimilated into the Bayesian structure; represent the states through which a network may transition into over time. The links between these nodes are transitions between the connected states that account for both neighbouring node dependencies and transition likelihoods - as weights in the form of a state transition matrix. The node dependencies are expressed as a set of conditional probabilities over their current and adjacent neighbouring states. Such a relational dependency fully describes a transition likelihood matrix built over time. The inference process is then an optimization over the set of posterior states which follows a probability distribution pre-determined by observed apriori state behaviors. The conditional is given as: $P(H|e)$ where H denotes the posterior hypothetical state and e is the apriori variable under observation. As a direct estimator, the strength of using Bayesian based models is that there is no need for actual ground truths to be known. Over a sufficiently large enough sample space, Bayesian based methods are capable of converging to actual ground truths of the network based on previously observed training samples from the structural sub-space. A summary of research into relational pattern temporality and evolution is given in Table 2.02.

Ref.	Methods	Contribution	Weakness
[132]	Bayesian Inference	• Temporal structural change behavior modeling. • Link predictive approaches. • Bayesian probabilistic models - Dynamic Bayesian Networks (DBN).	• Performance of Bayesian models deteriorates as poor choice of priors grows. • Slow convergence due to intensive computational demands on complexity.
[58]	Regression	• Focuses on the temporality in link strength and future evolutionary structural effects. • Comparison of similarities in feature vectors of existing node pair relations. • Feature combinations like the Adamic/Adar(AA), Common Neighbors(CN), Jaccard's Coefficient(JC), Preferential Attachment(PA) and Time Score(TA).	• Ignores multi-layered complexity of social networks. • Assumes a relationally flat homogeneous structure. • suffers from inaccuracies when applied to real-life heterogeneous networks.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[57]	Vector Auto-Regression	• Focuses on the temporality in link strength and future evolutionary structural effects. • Comparison of similarities in feature vectors of existing node pair relations. • Feature combinations like the Adamic/Adar(AA), Common Neighbors(CN), Jaccard's Coefficient(JC), Preferential Attachment(PA) and Time Score(TA).	• Ignores multi-layered complexity of social networks. • Assumes a relationally flat homogeneous structure. • suffers from inaccuracies when applied to real-life heterogeneous networks.
[59]	Link Prediction	• Focuses on the temporality in link strength and future evolutionary structural effects. • Comparison of similarities in feature vectors of existing node pair relations. • Feature combinations like the Adamic/Adar(AA), Common Neighbors(CN), Jaccard's Coefficient(JC), Preferential Attachment(PA) and Time Score(TA).	• Ignores multi-layered complexity of social networks. • Assumes a relationally flat homogeneous structure. • suffers from inaccuracies when applied to real-life heterogeneous networks.
[60]	Link Clustering	• Focuses on the temporality in link strength and future evolutionary structural effects. • Comparison of similarities in feature vectors of existing node pair relations. • Feature combinations like the Adamic/Adar(AA), Common Neighbors(CN), Jaccard's Coefficient(JC), Preferential Attachment(PA) and Time Score(TA).	• Ignores multi-layered complexity of social networks. • Assumes a relationally flat homogeneous structure. • suffers from inaccuracies when applied to real-life heterogeneous networks.

Table 2.02. Survey of temporal relational pattern identification and extraction

Other structural temporality based methods used to predict links proposed by Munasinghe et. al. in [57], [58], [59] and [60] focuses on the temporality in link strength and future evolutionary structural effects. The goals of the methods used in their paper is to find a model that predicts node pairs which contain a high likelihood to be linked in a future timeframe by comparing similarities in feature vectors of existing node pair relations. To achieve this, they train the supervised machine learning method using the set of feature vectors to find missing links between node pairs which have a high relational likelihood - structurally. In addition, they used a combination of features like the Adamic/Adar(AA), Common Neighbors(CN), Jaccard's Coefficient(JC), Preferential Attachment(PA) and Time Score(TA). A major drawback of

their method is that the authors have ignored the multi-layered complexity of social networks and assumed the overall relational pattern to be a flat structure. As such, it suffers from problems of inaccuracies when applied to real-life heterogeneous networks.

2.5.3 Machine Learning Approaches

Machine learning approaches are used to identify and extract information of interest from a given training bag of relational pattern samples [133], [134], [135]. Its aim is to logically associate correct identifiers with an observed relational pattern/s of interest. Extraction of these features are then performed when the identifiers deduced from actual pattern observations matches with identifiers learned from training pattern sets [136]. Social information networks contain deeply stratified layers of relational structures which have been uncovered from studies in stochastic approaches. Machine learning aims to deeply acquire knowledge about such patterns formed by complex social relationships in a hierarchical fashion, to facilitate accurate identification and extraction of rich relational features from heterogenity [56]. Some of the popular machine learning approaches include implementations of rule induction, statistical, spatial model, supervised and unsupervised training methods [134].

Hierarchical structural models are very closely correlated with belief networks [133]. A belief network is a directed acyclic graph (DAG) structure of belief states derived from a Bayes probabilistic model. It describes the mutual dependencies between latent variables and observables as a set of probabilistic relationships between each other [114]. As a measure of link reliability, a DAG provides very good stochastic inferences about link existence and relational strength. From a shallow architecture perspective, belief networks are often modeled after Restricted Boltzmann Machines (RBMs) [137]. An RBM is a bipartie graph of visible and hidden layers. It is an Artificial Neural Network (ANN) model which is capable of learning the probability distribution over a set of inputs from a training bag of samples [137], [138]. Described in Figure 2.05, the bottom layer represents the visible feature vector v while the top layer represents the hidden vector h. The matrix of

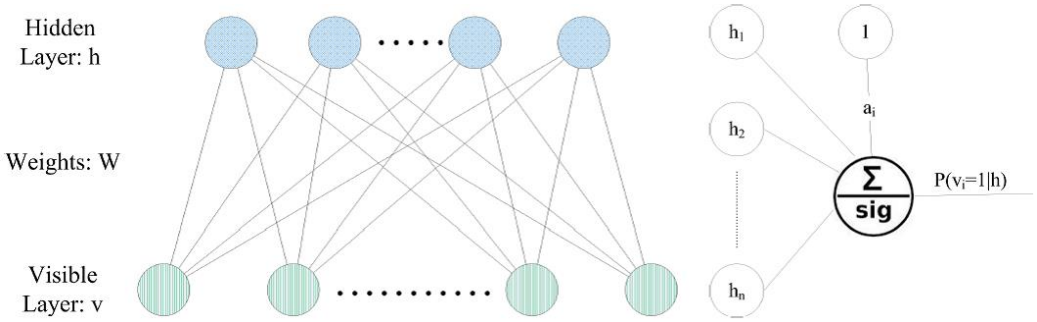


Figure 2.05. Structure of a Restricted Boltzmann Machine [138]. This diagram shows how nodes within a restricted boltzmann machine are connected in between layers and not within the layers themselves.

weights W contains the correlation of interaction terms between both visible and hidden units. RBMs are very efficient classifiers of structural behavior from hypothetical network beliefs [133]. In addition, they are also easy to train and converge to the distribution expectations very quickly. However, as a shallow implementation its performance is not capable of scaling up towards increasing complexities of growing networks [133], [134]. RBMs can be found in applications involving dimensionality reduction, clustering, collaborative filtering, feature learning and topic modelling tasks. As network grows and link prediction tasks become more complex, deeper generative models are often required to effectively learn the feature label classifiers from single step RBMs [139]. Hence, well-trained RBMs are often used to further train Deep Belief Networks (DBNs) for similar tasks within the context of high complexities [114]. Furthermore, RBM based DBNs are also capable of performing discriminative tasks of selecting belief networked states which can be used for link prediction or as a belief representation to the input of another classifier. Structural belief methods have been widely used in link predictive tasks of social and sentimental computing. Described in Table 2.03 are three pivotal studies from recent review, which have used machine learning (ML) and social theory (ST) models to solve for highly complex link identification and prediction tasks [113].

Ref.	Methods	Contribution	Weakness
[56]	Machine Learning	• Examines method of link prediction for two types of mobility data sets which contain location information. • Provides in depth analysis of machine learning methods like Decision Tree, Naive Bayes, Support Vector Machine, and Logistic Regression. • Investigates dynamic link prediction techniques.	• Little correlation between geographical displacement and how links are formed. • Excludes probabilistic likelihood ratios of existing links to structural importance.
[114]	Deep Belief Networks (DBNs)	• Establishes relationships of agreement and disagreement between actors as a signed social network (SSN). • Describes social behavourial patterns as derivatives to relations of mutual agreements. • Discriminative RBM based DBN models used for classification tasks in link prediction.	• Structural imbalance of link sign distributions skew classifiers towards larger probability density in the prediction tasks, giving rise to large errors. • Selective random removal of links to maintain balance between both positive and negative classes renders biased results. • Important structural information contained in the removed links are lost in the process. • Learning model is dependent on symmetry of observable distributions that scales poorly to asymptotic convergence of actual ground truths.
[111]	Social Theory	• Structural balance and social status theories to adequately determine the observation of edge signs. • Addresses the fundamental problem of inference using edge sign analysis. • New models for different node types to perform link sign prediction and ranking. • Defines 16 different node types and respective constraints - giving rise to different signed interconnecting edges (Figure 2.03).	• Uncovers relational trust reciprocity through the observation of different linked weights between nodes and their attached sentiments. • Ranks links through their respective sentiments reflected from their expressed edge signs. • Realiability and stability of "belief" structures are more readily achieved through the fusion of both node and triad features with the added robustness of Bayesian inference mechanisms.
[140]	Bayesian Learning	• Reconstructs accurate estimates of network property predictions through Bayesian learning inference mechanisms. • Develops a reliable framework to detect missing and spurious links in order to recover a stochastic reconstruction of structural "ground truths". • Describes link reliability as the probability of it's true existance given holistic observations of the hypergraph.	• Approach uses stochastic block models which are bounded by two fundamental ground truths. • Computationally prohibitive to integrate over all partitions. • Parameterization and tuning of candidate generating density (to be specified from a family of distributions) is a grey area.

Table 2.03. Survey of machine learning approaches

2.6 Classification And Prediction

Once structural patterns have been adequately identified and extracted, mid-level objectives of translating and understanding relational intelligence have been achieved. Data is then passed onto the next stage of the model for further analysis. Classification and prediction handles higher level interpretation processes within the relational pattern recognition framework [141]. Classification is a mid to higher level interpretation process which aims to form a knowledge base of abstractions from acquired and identified relational feature datasets [66]. The abstractions are mapped to labels which have been predetermined before the classification task. These labels can either be hard defined or trained from samples. The process of prediction further interprets these abstractions from classification tasks and makes inferences about future evolutions of observable relational pattern clusters [51]. Classification is a complex task which can be broken down into three broad sequence of steps:

1. Detection
2. Clustering
3. Labelling

The most prominent field of study in the area of OSNs which implement clustering methods for pattern classification is communitiy detection.

Community detection techniques have evolved from structural inferences over how nodes and links are grouped together [142]. Within the field of social science and networks, such scopes of study revolve around three broad areas of interest: Homogeneous Networks, Heterogeneous Networks and Social Internetworking Scenarios [143], [144], [145]. Homogeneous networks are often represented as a flat structure of relations amongst actors. They were originally developed to emulate structural mechanics of social networks from a single one dimensional perspective. As a mono-graphed structural model, they contain only single typed nodes and links [115]. Being the earliest model of social networks, their fundamental principles and mechanics have been extensively and fully developed. Some developments in this area include latent space models, block model representation, spectral clustering and modularity maximization.

2.6.1 Latent Space Models

In the latent space model, a low-dimensional Eucledian space is established as a global bounding condition to contain the arrangement of nodes linked together through a measure of "proximity" (minimal euclidean distance) between each other [106]. This "proximity" measure can be derived from any node-based social feature of interest (e.g. hometown, school, relatives, mutual friends, etc.). Typically, the construct of a proximity matrix offers the transformations of shortest paths within the euclidean manifold [146]. Once the nodes are arranged in their respective structure under this space, a popular clustering algorithm like the K-means approach is then applied on the framework to identify suitable clusters [147]. Clusters are formed from a grouping of relational patterns with high "proximity" measurements above a pre-defined acceptance threshold [148].

2.6.2 Block Models

In a block model approach, communities are determined from the corresponding edge densities between the interconnected nodes. It is based on the notion that nodes within a community are strongly related while nodes from different communites are weakly connected to each other [90]. "strong" and "weak" here refers to the number of edges seen within an observed "sliding window" timeframe. The stochastic approximation is given by the block matrix as: $S = \{0, 1\}$ - an $n \times k$ matrix. The approximation is discrete, and the solution to a probabilisitc block order matrix is often associated to the general case as being NP hard [79]. Where N refers to the number of nodes and P to the edge probabilities of a symmetric community partition. A common approach to reduce complexity of the solution is to relax the block matrix S so that it appears more continuous and will also satisfy certain orthogonal constraints: $S = IK$ [90]. This step further reduces the solution to finding the optimization of S such that only the top K eigenvectors with maximum eigenvalues are uncovered from the process. Similar to the latent space models, a k-means clustering algorithm can then be used to easily recover the expected community blocks of top K eigenvectors forming the desired relational pattern which can be classified.

2.6.3 Spectral Clustering Models

Spectral clustering is a method developed to "divide" the supergraph into sub-graphs through the use of a singular "cut" metric [149]. This is derived from a spectrum of structural and node similarities which attempts to reduce network dimension. The cut metric is defined as the number of edges between two disjoint set of nodes through which a disection of the network is to occur [75]. Studies on this method seek to minimize the number of distant interconnecting edges between the two different communities [150]. The spectral approach on its own works well in situations where sparse graphs are considered. However, as more nodes and connections are added to the homogeneous topologies, spectral algorithms become increasingly hard to solve [151]. A tricky approach which spectral clustering adopts is the use of nonlinear dimensionality reduction techniques (e.g. information embedding(LLE), Isomap, Laplacian eigenmaps, Manifold Alignment, Diffusion maps, etc.) to represent the high dimensional data graphs in a more "compact" and "flat" structure that becomes more computationally managable [152]. However, a drawback is that dimensionality reduction functions to eliminate features and information which may be correlated to the relational patterns of a community structure. Another tricky parameter which needs to be defined for spectral clustering to work effectively in a structural graph context is also known as the "quality threshold" of a cluster. This threshold defines the "strength" of a connected cluster (i.e. how densely connected the intra-community nodes are - in relation to inter-community vertices) [152]. If this threshold is set too high, then multiple small and densely connected clusters will be partitioned, but the larger loosely connected ones will be overlooked. If instead, this threshold is defined too low, then small numbers of huge clusters (oftentimes merged clusters) will be partitioned and the smaller clusters left out. Both thresholding extremes which community detection methods employ using spectral mechanics are prone to erronously detect the exact community structure of relational patterns from ground truths [151].

2.6.4 Modularity Measure

Modularity is defined to measure the structural strength of a graph partition (which is usually derived from spectral methods mentioned above) [90]. The

structural "strength" feature in this measurement is obtained from node degree distributions (Figure 2.07) between nodes of the same community. For a graph of n nodes and m connections, the expected number of edges between two nodes A and B in any particular consideration is:

$$E_{A,B} = \frac{d_A d_B}{2m} \tag{2.17}$$

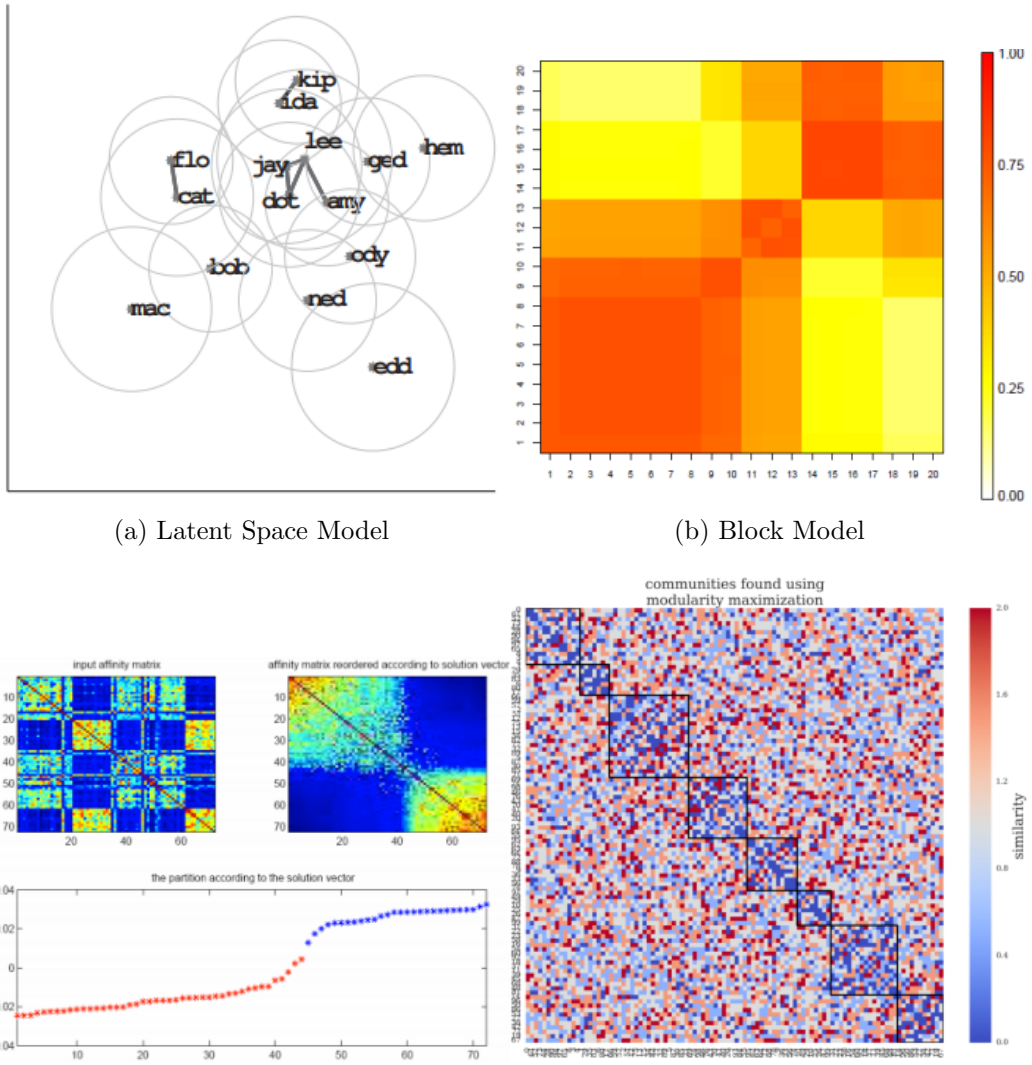
The expectation is a stochastic approximation from a random number of connections taken from ground truths. Hence, the difference between actual number of links and approximated expectations is given as the deviation defined by:

$$R_{A,B} - E_{A,B} \tag{2.18}$$

where $R_{A,B}$ is the number of actual links counted from the graph. Logically, the problem set reduces to optimizing $R_{A,B}$ such that $R_{A,B} - E_{A,B}$ is maximized over all possible nodes within the (community) locality of the search space. This gives the modularity Q which can be expressed as:

$$Q = \frac{1}{2m} \sum_{AB} \{R_{A,B} - E_{A,B}\} \frac{S_A S_B + 1}{2} \tag{2.19}$$

where S_X is defined as the membership variable which takes on either one of 2 binary variable representations (1 or -1). if node A belongs to the parent community, then $S_A = 1$ otherwise, if node A belongs to the neighbouring community, then $S_A = -1$. Modularity methods however, suffer from resolution limits especially when employed within large graph models [90]. This is because the stochastic expectation (often refered to in literature as the random null model) implictly assumes that there are closed paths between any two arbitrary nodes picked out for consideration under this model [79]. This means that as the network grows aribitrarily large, m (the number of links within the network) increases exponentially. Hence, the expectation reduces proportionately by an exponential amount. This reduction occurs such that when there is a single defining link between any 2 nodes in the graph, a high modulartiy is calculated. This in turn leads to multiple clusters merging up as a huge single community structure of nodes and links alike. Thus, optimizing modularity within a large network structure will fail to detect small communities whose expectation of community (modularity)



(a) Latent Space Model

(b) Block Model

(c) Spectral Clustering

(d) Modularity Maximization

Figure 2.06. Four most popular approaches to the Community Detection Problem [149] [143] [144] [115]. This diagram shows the four widely researched methods on community detection - using the Latent Space Model, Block Model, Spectral Clustering and Modularity Maximization.

"strength" remains the same - invariant to superstructural "growth" in volume, dimensionality or complexity.

The problem of community recovery has recently gained a renewed surge of interest in the academic society and has been intensely studied in statistical mathematics [7], [106] & [147], computer science (where it is known as the planted partition problem) [75], [153] & [154], and theoretical physics [155], [156] & [53]. The block model representation is often used in its sim-

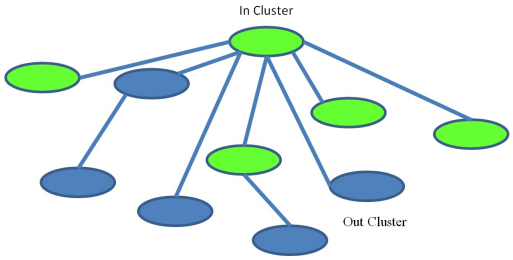


Figure 2.07. Typical Node degree arrangement of a mixed cluster network. [143] This diagram shows how nodes with high degrees are interpreted as having high membership scores to one cluster and low membership scores to another adjacent cluster.

plest from in this extent to represent a set of n vertices, partitioned into two or more clusters with edge probabilities $\frac{a_i}{n}$ where i refers to the specific cluster in a conditional relation. The goal is to reconstruct the underlying clusters of relational patterns from observations of symmetric differences of the graph. The community recovery threshold can be extended across multiple communities and up until recently, multilayer heterogeneous relational pattern clusters as well. As an example, for dual equal sized community sub-structures co-existent within the superstructure of a social graph, the threshold of recovery can be expressed conditionally as:

$$\frac{n(p-q)^2}{2(p+q)} > 1 \tag{2.20}$$

The maximum fraction of recovery is:

$$\lim_{n\to\infty} E[\varphi(\hat{\sigma},\sigma)]_{\partial}^{inf} = Q(\sqrt{\bar{v}}) \tag{2.21}$$

Where infimum ranges over all possible estimators $\hat{\sigma}$ based on graph G such that it classifies communities correctly at a maximum likelihood of $Q(\sqrt{\bar{v}})$. $\varphi(\hat{\sigma},\sigma)$ is the fractional misclassification of community vertices based on the estimator $\hat{\sigma}$ of graph G given by:

$$\varphi(\hat{\sigma},\sigma) = \frac{1}{n}\min\{\sum_{i=1}^{n} 1_{\{\sigma_i\neq\hat{\sigma}_i\}}, \sum_{i=1}^{n} 1_{\{\sigma_i=\hat{\sigma}_i\}}\} \tag{2.22}$$

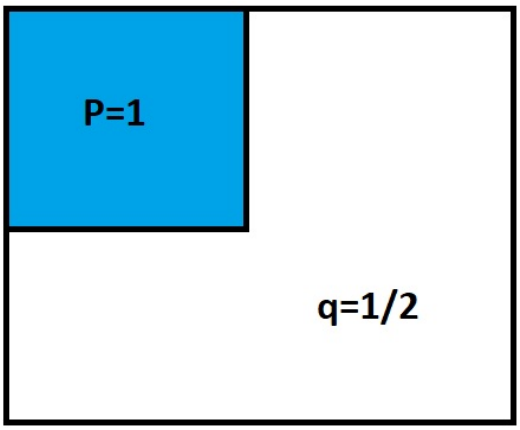


Figure 2.08. Stochastic Block Model of a Planted Clique. This diagram shows how logical definitions of the planted clique problem is formulated statistically [95].

The threshold for weak recovery is then defined as:

$$\frac{n(p-q)^2}{2(p+q)} \to \infty \tag{2.23}$$

The corresponding threshold for exact recovery is given as:

$$p = \frac{a \log n}{n}, q = \frac{b \log n}{n} \forall \sqrt{a} - \sqrt{b} \geq \sqrt{2} \tag{2.24}$$

For linear community size $K = \rho n$, there exists a polynomial time function f which defines the threshold of exact recovery as:

$$\rho f(a,b) > 1 \tag{2.25}$$

So that exact recovery is perfectly impossible if,

$$\rho f(a,b) < 1 \tag{2.26}$$

In addition to community recovery thresholds, there are computational limits of recovery, which refers to the planted clique problem.

In discrete community detection instances, this can be defined as a computational gap. In contrast however, there are no computational gaps for polynomial time algorithms. For an exact recovery using a Maximum Like-

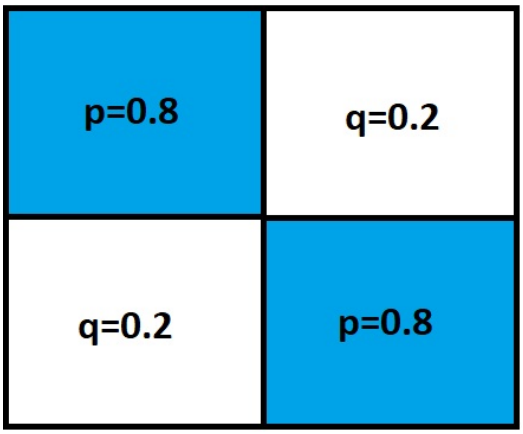


Figure 2.09. Stochastic Block Model of Equal Sized Stub Community Structures [95]. This diagram shows how logical definitions of equal sized communities are formulated statistically under the planted clique problem.

lihood (ML) estimator,

$$K \geq 2(1+\epsilon)\log_2 n \tag{2.27}$$

In polynomial time series, exact recovery is attainable by:

$$K \geq \epsilon\sqrt{n} \tag{2.28}$$

Finally, for a random network, exact recovery is believed to be NP-hard and is given as:

$$K = o(\sqrt{n}) \tag{2.29}$$

Yet, as with all community detection techniques used in todays context of social networks, it still suffers from a resolution limit. The resolution limit of any defining estimator $\cap\sigma$ set to work on a graph G is defined as the smallest sized communities (also termed as cliques) of a number of nodes n. Below which, for $2\log_2 n < 40$ and $\sqrt{n} = 1000$, such a community Z cannot be detected. The computational gap for the small community Z is given as:

$$K \in [(2+\epsilon)\log_2 n, o(\sqrt{n})] \tag{2.30}$$

Several methods have been proposed to overcome the problem of both information theoretic and computational limits of recovery that have since in recent years become a major problem when dealing with heterogeneity of relational patterns in clusters. A summary of research contributions is given in Table 2.04.

Ref.	Methods	Contribution	Weakness
[157]	Multi-View Clustering	• Multi-View learning enables mapping of functions. • Arrangement of different structural patterns from various data spaces together. • Discovery of optimal alignment for learning.	• Performance deteriorates when community node size decreases below 10^{12}. • Will have to revert to supervised or semi-supervised techniques below 10^{6} node sized sub-communities.
[158]	Linked Matrix Factorization (LMF)	• Exploits multiple sources of information. • Makes better inferences on entities through clustering. • Cluster optimization of fused unsupervised and semi-supervised graphs.	• Reliability of detected communities unquantified. • Correlations between multiple views not considered during clustering.
[159]	Graph Kernel Min-Max Optimization	• Draws a compromise between supervised and unsupervised learning approaches • Optimization of graph constructs. • Min-Max solution for large graphs.	• Regularizes over kernel manifolds to conserve node labels. • Susceptible to the resolution problem of community detection approaches.
[94]	Evolutionary Clustering	• Addresses the evolving nature of HNs. • Temporal information-aware spectral clustering framework. • Cluster optimization through changes in structural temporality.	• Potential for large error growth within latent community structures. • Sensitive to manually tuned parameterizations. • Slow convergence.
[160]	Cluster Prediction	• Edge-centric clustering scheme. • Detects sparsification at community edge localities. • Discriminative learning through social features.	• High performance sensitivity to increasing social dimensions. • Dimensionality threshold needs to be tuned for optimization. • Scales poorly to increasing complexity with low dimensionality sparsifications. • Contains polynomial time gaps where algorithms break down.
[161]	Modularity Maximization	• Community based ground truth predicates. • Graph sparsification of dense sub-nodes and links. • Soft weight node community membership assignment.	• Performance degrades when network sparsification is low.

Table 2.04. Survey of community detection and pattern classification

2.7 Learning And OAI

Learning is an iterative core process through which accurate and reliable interpretation of relational pattern abstractions can be easily recognized [162]. Computational Cognitive Artificial Intelligence (CCAI) has been the hallmark of many research developments in the field of applied computer science and engineering topics [163]. Some important applications of study include detection, classification, outlier modeling, tracking, recognition, prediction, navigation, autonomy, etc. Generally, AI algorithms are becoming a trending approach to solving many open ended problems of real world models where closed form solutions are becoming increasingly complex to derive [164], [165], [166]. The key difference between developed AI and statistical methods are that the latter relies on hard conditions whereas the former softens these thresholds for similar computational intelligence tasks [167], [168]. As a consequence, statistical techniques require that the dataset of study conforms to an Identical Independent Distribution (I.I.D). Within this characteristic spread pattern, a closed form solution can be quite easily acquired. By contrast however, AI approaches are very extensible in dealing with non-I.I.D datasets, dis-symmetry and asymptotic convergence. All of which, are serious limitations of Statistical Relational Learning (SRL) techniques [169], [170]. A typical example of this is the wide use of the "fuzzy" concept when performing activities like classification, ranking, prediction, etc. This framework is best illustrated as a translation of discrete to continuous probabilistic distributions of community memberships during classification tasks as in Figures 2.10 and 2.11.

OAI refers to a subset of CAI where traditional learning algorithms which were applied to both statistical and fuzzy set representations have evolved to handle the dynamism of online data [19]. These methods handle similar CAI tasks in a challenging real-time online setting. A summary of key developments in the field of OAI is given in Table 2.05.

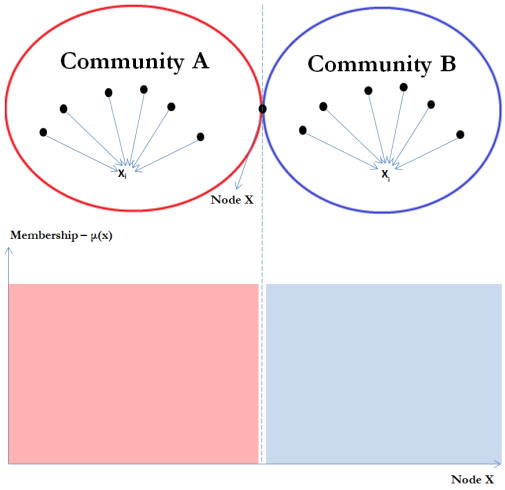


Figure 2.10. A statistical Set representation. This diagram shows the logical representation of a crisp set where membership ambiguities exist at set boundaries.

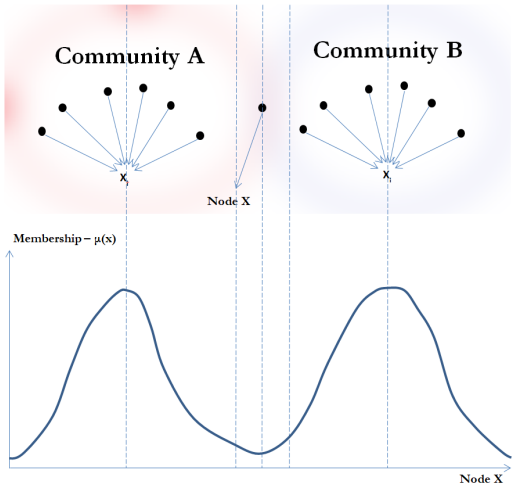


Figure 2.11. A Fuzzy Set representation. This diagram shows the logical representation of a fuzzy set where membership scores are measured as probabilities instead of binary “1” and “0”.

AI	Type	Algorithm	Description
AR(p) & VAR	Time Series Predictive	• Characteristic equation: $\phi_t = G + \sum_{i=1}^{\rho} \Lambda_i \phi_{t-i} + \epsilon_t$. • Moving Summation: $\delta[C]\phi_t = d + \epsilon_t$. • Characteristic Polynomial: $\varsigma(\tau) = \sum_{\nu=1}^{x} a_\nu y_\nu^{-\lvert\tau\rvert}$.	• Linear output to input mapping of stochastic differences. • Structured representation of random I.I.D. stochastic processes. • Simplistic and elegant approach to small world problem sets. • Solution complexity of O(mn).
SVM	Linear least squares classifier	• Decision Boundary: $\vec{w}^t \cdot \vec{u}^t = \vartheta$. • Hard-Margin Objective Function: $\xi = max[0, (\overrightarrow{\chi_+} - \overrightarrow{\chi_-}) \cdot \frac{\vec{u}}{\lvert\vec{w}\rvert}] = min[\frac{1}{2}\lvert\lvert\vec{w}\rvert\rvert^2]$ • Soft Margin Objective Function: $min[\frac{1}{z}\sum_{i=1}^{z} max(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b))] + \varrho\lvert\lvert\vec{w}\rvert\rvert^2$ • Non-Linear Kernel mask: $\vec{w} = \sum_i \alpha_i y_i \psi(\vec{x}_i)$ where $\psi()$ is the kernel transformation of the input data space.	• Linear "Widest street approach" classification technique. • High classification accuracies across many sample sized applications. • Relies heavily on limited kernels for non-linear classifications. • Adapts poorly to similar tasks for large complex datasets (hyperplanes). • Solution complexity scales to $O(\alpha K^d)$. Where K is the kernel displacement and d is the plane dimensionality. For a uni-dimensional plane, complexity reduces to a linear optimization of input summations.
Fuzzy Logic	Soft Boundary classifier	• Hard Membership Functions: $\Pi(\mu_{A/B}) = \begin{cases} 0 & \text{Where } \mu_{A/B} \leq \mu_{A/B_b} \\ 1 & \text{Where } \mu_{A/B} \geq \mu_{A/B_b} \end{cases}$. • Soft Membership Functions: $\mu_{\bar{A}} : X \to [0, 1]$ and $\mu_{\bar{B}} : X \to [0, 1]$.	• Substitutes decision ambiguities at boundaries with probabilistic inference. • Trades off deterministic measures for discriminative capabilities at distribution boundaries. • Models a distributive spread of data classifier probabilities . • Adaptable to most real world data problem sets and features. • Min-max approach to boundary constraints. • Solution complexity scales to $O(m^n)$. Where m is the number of fuzzy boundary decisions and n is the control loop number.

Continued on next page

—*Continued from previous page*

AI	Type	Algorithm	Description
K-NN	Proximity discriminator	• Euclidean Distance: $\sqrt{\sum_{i=i}^{k}(x_i - y_i)^2}$ • Manhattan Distance: $\sum_{i=1}^{k} \|x_i - yi\|$ • Minkowski Distance: $[\sum_{i=1}^{k}(\|x_i - yi\|)^q]^{\frac{1}{q}}$ • Hamming Distance: $D_H = \sum_{i=1}^{k} \|x_i - y_i\|$ • Conditional Probability Estimation: $P(y = j\|X = x) = \frac{1}{K}\sum_{i \in A} I(y^(i) = j)$	• Similarity measure based benchmark classifier. • Supervised learning predictor. • Non parametric Instance-based learning algorithm. • Majority confidence voting. • Linear complexity $O(mn)$ where m is the target node and n is k-nearest neighbor.
Bayesian Learning	Stochastic convergence and divergence	• Bayes Rule: $P(X\|Y) = \frac{P(Y\|X)P(X)}{P(Y)}$ • Posterior Distribution: $P(\gamma\|X, \rho) = \frac{P(X\|\gamma)P(\gamma\|\rho)}{P(X\|\rho)}$ • Bayesian Inference: $P(\tilde{x}\|X, \rho) = \int P(\tilde{x}\|\gamma)P(\gamma\|X, \rho)d\gamma$ • MCMC Expectation: $E_P[G(\theta)] \approxeq \frac{1}{s}\sum_{n=1}^{S} G(\theta_i)$ • Gibbs Sampling: $E(\theta) = P(\theta_i\|\theta_{j \neq i})$ • HMC form: $P(x, v) \propto e^{-E(x)}e^{\frac{-v^T v)}{2}} = e^{-H(x,v)}$ • HMM: $P(X_{1:T}, Y_{1:T}) = P(X_1)P(Y_1\|X_1)\prod_{t=2}^{T} P(X_t\|X_{t-1})P(Y_t\|X_t)$	• Addresses high complexity tasks which involve high level inferences. • Trades versertility of complex predictive task handling for speed. • Adaptable to information sparsity in a dataset. • Hamiltonian dynamics eliminates diffusive behavior of simple random-walk processes. • HMMs stochastically models unobserved states as a markov process.
Entropy	Information randomness	• Maximum Entropy: $H_{max}(p) = max[0, -\sum_{x \in \varepsilon} p(x)logp(x)]$ where $p(a, b)$ is the target distribution, $x(a, b), a \in A, b \in B$, and $\varepsilon = A \times B$ • Minimum Entropy: $H_{min}(G) = min_{x \in (G)} log_2(\frac{1}{Pr[x]})$	• Defines likelihoods of source information distribution at the receiver. • Encodes and decodes randomness into and from information which makes it indiscernible to third parties. • Scales exponentially with increasing complexity.

Continued on next page

—*Continued from previous page*

AI	Type	Algorithm	Description
Adaptive Learning	Dynamic data modeling	• Multiple Instance Active Learning (MIAL): $f_s(B) = \frac{1}{L}\sum_{t=1}^{L}\frac{1}{N_t}\sum_{i=1}^{N_t} y_{l,i}T_r(B, B_{l,i})$ where $f_s(B)$ is the discriminative classifier, $T_r(B, B_{l,i})$ is the transfer function from source bag category $B_{l,i}$ to target bag category B instances. • Term Frequency-Inverse Document Frequency (TF-IDF): $tf_{t,d} \cdot idf_{t,d} = \frac{f_{t,d}}{\sum_{t=0}^{T} f_{t,d}} \times log\frac{N_d}{n_{d,t}}$ • Passive-Aggressive Learning: $\tau_t = minC, \frac{l_t}{\|x_t\|^2}$(Linear Form), $\tau_t = \frac{l_t}{\|x_t\|^2+\frac{1}{2C}}$(Online Form). Where C denotes the "aggressiveness" of each update step, l_t specifies the hindge losses with respect to time t and x_t refer to the data classifier instances in t. • Kernel-Based Learning: $k(G, G') = \sum_{k=0}^{\infty}\mu(k)q_\theta^T W_\theta^T p_\theta$, Where $k(G, G')$ is the transform mask, $\mu(k)$ is the kernel mean, q_θ^T is the target probability distribution, p_θ is the sample probability distribution and W_θ^T represents the weighted score of random walks within the kernel function k.	• Provides greater generalization across multiple data samples. • Target distribution scales linearly with input. • Reduced time complexity in performing similar tasks. • Heavy reliance on kernel distribution masks. • Structured learning approach.
Ensemble methods	Clustering and classification	• Clustering Ensemble: $\phi(F, \Gamma) = F(P_1, P_2, P_3, ..., P_m) = F(\Gamma)$ where $\phi(F, \Gamma)$ is the clustering ensemble built with the consensus function F and an m-partition ensemble $\Gamma = [P_1, P_2, P_3, ..., P_m]$. • Adaptive Clustering Ensemble (ACE): $S_c(C_{j\alpha}^q, C_{jl}^l) \geq \alpha$ (merging condition) $S_c(C_{j\alpha}^q, C_{jl}^l) \leq \alpha$ (dissolution condition)	• Parallel set of individual models for machine learning task. • Concatenation of multiple clustering models (partitions) into a single consolidated partition. • Requires complicated aggregating strategies for combining clusters generated by individual clustering models. • Complexity of methods scales poorly with increasing data dimensionality. • Trades accuracy for performance of classification tasks.
Matrix Factorization	Data Reduction	• LU Decomposition: $PAQ = LU$ where L is the lower triangular matrix, U is the upper triangular matrix, P is the Pivot matrix and Q is the permutation matrix. • QR Decomposition: $A = QR$ where Q is the orthogonal matrix and R is the upper triangular matrix. • Cholesky Decomposition: $A = LL*$ where L is the lower triangular matrix, $L*$ is the conjugate transpose of L.	• Dimensionality reduction to form decompositions which are easily handled by most Machine Learning algorithms. • Linear decompositions preserves feature reducability. • Data irreducability increases decomposition cost. • Unable to handle reducable decompositions of non-IID data sets. • Unsuitable for non-linear reduced data mappings.

Continued on next page

—*Continued from previous page*

AI	Type	Algorithm	Description
Tensor Flow	Multi-dimensional learning	• Multi-Dimensional Classification (MDC): $y^* = argmax_y p(y\|x^*)$. Where x is the input and y is the output and $p(y\|x^*)$ is the label expectation of the output against a given input sample x^*. • Probabilistic Classifier Chains (PCC): $\hat{y}^* = hx^*\|s = argmax_{\hat{y}} p(\hat{y}\|x^*, s)$. Where $p(\hat{y}\|x^*, s)$ is the chained likelihood of output to input training sample label instances $p(\hat{y}\|x^*)$ against and expectation index of labels s. • The penalty / reward function: $J(s) = \sum_{i=1}^{N} p(y^(i)\|x^(i), s)$. Where the payoff $J(s)$ is described as the optimal label order $p(y^(i)\|x^(i), s)$ during the learning phase.	• Efficient exploration / learning of label-sequence space. • Models dependencies between classifiers effectively. • Multiple instance / dimensional learning increases computational cost. • Error propagation down the chain of learned labels are hard to control. • Improves classifier and cluster predictions.
ANN	Neural Net Learning Models	• Restricted Boltzmann Machine (RBM): $y_j = \sum_{i=0,j=0}^{m,n} W_{ij} x_i h_j + b$. Where y_j is the j-array restricted layer vector, x_i is the i-array input layer vector, h_j is the hidden layer vector and b is the bias. • Extreme Learning Machine (ELM): $y_n(x) = \sum_{i=1}^{n} \beta_i h_i(x)$. Where $y_n(x)$ is the output vector of n hidden nodes from input x, $h_i(x)$ is the hidden layer output function $H(a_i, b_i, x)$ and β_i is the hidden layer weight of the ith output node. • Multi-Layer Perceptron: $y_j = \sum_{i=0,j=0}^{m,n} \sigma_{ij} W_{ij} x_i + b$. Where σ_{ij} is the logistic activation function. The output y_j of each layer becomes the input x_i the the next layer above. • Deep Neural Network: $y_n = U^T H_n$ where y_n is the output vector, U^T is the upper layer weight matrix and H_n is the hidden layer activity matrix.	• Scales adequately to large complex problems. • Stacks multiple layers of learning classifiers to solve highly complex relations . • Individual layers are made up of simple classifier belief states (neurons) and activation weights which are learned iteratively. • Computational efficiency of hidden variables are preserved during the learning phase. • Data overfitting and underfitting is addressible by generative pretraining steps.

Table 2.05. Survey of key developments in OAI

This section gives a brief overview on three main OAI approaches that have been widely developed and implemented in the online scene. Their objectives tackle large scale sentimental and affective computing of continuously evolving social transaction patterns.

2.7.1 Learning On-The-Fly

Active Online Learning (AOL) is a branch of machine intelligence which seeks to resolve predictions in real time through continuous sequential inputs from data streams [19]. Generally, from a statistical viewpoint, Machine Learning (ML) can be categorized into either Frequentist or Bayesian views. Bayesian inferences are often performed with probabilistic distributions whereas Frequentist inference is done through fixed parameters of random data samples [171]. The Bayesian inference mechanism is a soft approach to the hard variable constraints of Frequentist inferences. For example, in Frequentist based methods, unknowns are often fixed to a hard value constraint, whereas in Bayesian based techniques, unknown parameters are often modeled as probabilistic kernels [133]. In ML, the following learning paradiams can be established:

- Supervised Learning (SL)
- Unsupervised Learning (UL)
- Semi-Supervised Learning (SSL)
- Reinforcement Learning (RL)
- Active Learning (AL)

The objective of online learning is similar to other ML tasks. It seeks to minimize prediction errors made between estimations and observables [172]. The key differences between this domain of ML techniques and other offline ML approaches is given in Table 2.5

AOL techniques can be viewed as a more generalized model of the Recursive Neural Network (RNN). In similarity, both methods rely on new instances of data arrivals and sequential feedback from past predictions as inputs into the model. The key difference which seperates AOL techniques (e.g. passive-aggressive learning) to RNN approaches is that the sequenced feedback of an RNN is very well defined and constrained to form a predictive

Table 2.5: Table of differences between Online and Offline ML approaches

Online ML	**Offline ML**
AOL methods rely on a continuous sequantial stream of data to approximate a ground truth in an online setting.	Offline ML techniques rely on training data (often broken up into batches) of past event occurances to learn a target function within a test set.
AOL requires extremely fast convergence to prediction expectations in order to deal with temporal shifts in data (a.k.a. topic drifts).	Offline ML have vast resources of computational power and time at their dispense to converge to certain expectations unique to the task in question.
Due to a strict requirement of fast convergence to predictive expectations, AOL models often adopt a shallow design at best.	In contrast, offline ML are capable of extending their models across multiple layers to deeply learn sophisticated features within large datasets.
Shallow ML architectures adopted by AOL techniques lack the representational power of more complex metadata.	In contrast, deep architectures extended by offline ML approaches are very powerful with representational aspects of high dimensional data.
AOL structures rarely encounter such problems with gradient descent because of their relatively "flat" architectures.	Offline ML architectures - because of their complicated structure designs are also very hard to train and often runs into well known convergence issues (e.g. instability, ill-positioning, vanishing and exploding gradients).
As a result, AOL models are relatively easy to train and are agile enough to update itself instantly and efficiently as new live data streams arrive.	The drawback of deep offline ML architectures however, are that they require extensive re-training effort for every new learning target objective instance.

feedback pattern, whereas in AOL the algorithm constantly deals with feedback that is non-predictive and unpatternized due to data temporal shifts [173], [174], [67]. Consequantly, topic drifts are a major issue which AOL algorithms struggle with when trying to make accuracte short-term predictions. Generally speaking, the AOL domain derives its key characteristics from three main active learning architectures: Active Agressive Learning (AAL), Markov Decision Process (MDP) - A.K.A. Reinforcement learning and Kernel-Based Learning [83].

AAL architectures rely heavily on constant iterative updates of predictive change data. A key characteristic which AAL models exhibit are the lagging effects of persistant updates - even in the event of missing data inputs [175], [83]. In situations when the stream of fresh inputs are predictable and continuous with no breaks in between, Selective Sampling is often used to draw sets from a fixed distribution manifold. However, when there are discontunities in the streaming data, sample sets can be generated stochastically. The two settings used in Active Aggressive Learning structures are often termed as Selective Sampling (SS) and Label Efficient (LE) respectively [175], [176], [171]. Generally, SS approaches work best under well-constrained environments where the data stream is structured and stable. Common key objective functions of SS techniques revolve around optimizing prediction accuracy of test datasets. In contrast, LE techniques are adaptable to unstructured update environments where data stream is compromised in some manner (e.g. inconsistant streaming input, missing data, etc.). To make up for missing sampled inputs, LE are tasked with representing these sparse instances with posterior inferences from previous sample observables. Thus, key LE objective functions instead, revolve around optimizing representational accuracy used in the predictive process of test datasets [177], [178]. In summary, SS methods are estimator focused algorithms where persistance updates are used to minimize predictive losses, while LE techniques are approximator focused algorithms where persistance updates are used to maximize probabilitiy densities of missing distributions.

A summary of key SS algorithms and their objective functions are given in Table 2.06.

Algorithm	Objective Function
Margin-based Selective Sampling Algorithm	• Characteristic label prediction rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Cumulative Loss / Regret: $\sum_{t=1}^{T}[Pr(y_t w_t^T x_t \leq 0) - Pr(y_t w^T x_t \leq 0)] \leq N + \mathbb{C}L + 4lnT$. • Model Complexity: $C = \mathbb{C}L + O(\frac{d+lnT}{\lambda^2})$.
Bound on Bias Query	• Characteristic label prediction rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Cumulative Loss / Regret: $\sum_{t=1}^{T}[Pr(y_t w_t^T x_t \leq 0) - Pr(y_t w^T x_t \leq 0)] \leq \min_{\epsilon \in [0,1]}(\epsilon T_\epsilon + (2+e)[\frac{1}{k}](\frac{8}{\epsilon^2})^{\frac{1}{k}} + (1+\frac{2}{\epsilon})\frac{8d}{\epsilon^2} ln(1 + \frac{\sum_{t=1}^{T} Z_t}{d}))$. • Model Complexity: $C = O(dT^k lnT)$.
Parametric BBQ	• Characteristic label prediction rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Cumulative Loss / Regret: $Pr(\lvert w_t^T x_t - w^T x_t \rvert \leq \epsilon) \geq 1 - \delta$. • Model Complexity: $C = O(\frac{d}{\epsilon^2}(ln\frac{T}{\delta}) ln\frac{ln(\frac{T}{\delta})}{\epsilon})$.
Dekel Gentile Sridharan (DGS)	• Characteristic label prediction rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Cumulative Loss / Regret: $\sum_{t=1}^{T}[Pr(y_t w_t^T x_t \leq 0) - Pr(y_t w^T x_t \leq 0)] \leq \inf_{\epsilon > 0}[\epsilon T_\epsilon + O(\frac{dlnT + ln(\frac{T}{\delta})}{\epsilon})]$. • Model Complexity: $C = \inf_{\epsilon > 0}[T_\epsilon + O(\frac{d^2 ln^2(\frac{T}{\delta})}{\epsilon^2})]$.

Table 2.06. Survey of selective sampling algorithms in OAL

A summary of key LE algorithms and their objective functions are given in Table 2.07.

Algorithm	Objective Function
LE-Perceptron	• Characteristic update rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Update Rule: $w_{t+1} = w_t + y_t x_t$ • Cumulative Error Function: $\mathbb{C}[\sum_{t=1}^{T} M_t] \leq (1 + \frac{R^2}{2\delta})\frac{\bar{L}_{\gamma,T}(w)}{\gamma} + \frac{\lvert\lvert w^2 \rvert\rvert (2\delta + R^2)^2}{8\delta\gamma^2}$.
LE Second-Order Perceptron	• Characteristic label prediction rule: $\hat{y}_t = sgn(\hat{p}_t), \hat{p}_t where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Update Rule: $u_{t+1} = u_t + y_t x_t$ and $A_{t+1} = A_t + x_t x_t^T$. • Cumulative Error Function: $\mathbb{C}[\sum_{t=1}^{T} M_t] \leq \frac{\bar{L}_{\gamma,T}(w)}{\gamma} + \frac{\delta}{2\gamma^2} w^T \mathbb{C}[A_T] w + \frac{1}{2\delta}\sum_{i=1}^{d} \mathbb{C} ln(1 + \lambda_i)$.

Continued on next page

—*Continued from previous page*

Algorithm	**Objective Function**
Adaptive LE Second-Order Perceptron	• Characteristic label prediction rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Update Rule: $u_{t+1} = u_t + y_t x_t$ and $A_{t+1} = A_t + x_t x_t^T$ • Cumulative Error Function: $\mathbb{E}[\sum_{t=1}^{T} M_t] \leq \frac{L_{\gamma,T}(w)}{\gamma} + \frac{\delta}{2\gamma^2} w^T \mathbb{E}[A_T] w + \frac{1}{2\delta} \sum_{i=1}^{d} \mathbb{E} ln(1+\lambda_i)$.
Passive Aggressive Active (PAA) Learning	• Characteristic label prediction rule: $\hat{y}_t = sgn(p_t) where p_t = w_t^T x_t$. Where W_t is the weighted data correlation between input x_t and predictive output p_t. • Update Rule: $w_{t+1} \leftarrow w_t + \tau_t y_t x_t$ • Cumulative Error Function: $\mathbb{E}[\sum_{t=1}^{T} M_t] \leq \beta(\frac{\delta+1}{2})^2 ||w||^2 + (\delta+1) C \mathbb{E}[\sum_{t=1}^{T} Z_t l_t(w)]$.

Table 2.07. Survey of label efficient algorithms in OAL

Reinforcement Learning (RL) is a behavioral learning method which adopts a risk reward approach [179], [167], [180]. In a stochastic state environment, the learner (agent) is concerned with taking risks in the environment which maximize rewards. Relationally, the environment can be modeled as a state machine containing three main characteristics:

1. State Transition Function
2. Observation (Output) Function
3. Rewards Function

The learner similarly can also be modeled as a state machine which operates under that environment. Learners are usually characterized by two main attributes:

1. Risk Function
2. Rewards Function

The learner undertakes risks by transitioning from state to state within the environment to achieve a reward at the output for arriving at expectations which are close to the environment's observation. The smaller the errors are in the learner's risk approximation, the larger their corresponding rewards will be. The goal of the learner therefore is to discover an optimized risk to policy/output function through feedback and updates [181].

RL is often used as an unsupervised Markovian "chain-of-states" learning mechanism to discover an optimal "walk" from input to output which minimizes the risk taken along its path while maximising the rewards achieved at

the end (output) [35]. While intuitively powerful in its approach at solving problems involving sequential dynamics and optimization of similar objective functions, they scale poorly with time - especially if the manifold is particularly large [182]. Furthermore, the "randomness" of the discovery / risk function means that experiments done in one instance may not be repeatable in another instance - although risk rewards ratio in all instances may approximate very closely to each other [179]. OAL adopts the partial randomness of RL approaches by balancing tradeoffs between exploration (of uncertainty) and exploitation (of apriori knowledge).

Kernels are essentially masks used to transform the characteristics of a learner from a singular (usually linear) form to multiple (non-linear) framework [175], [82], [83]. For example, a traditional SVM approach can only be used on datasets which are linearly homogeneous. Such a stochastic property is being reflected as having an Identically, Independent Distribution (I.I.D) [183]. IID datasets are deterministic to a wide variety of solvers for arriving at accurate estimations of unknown observables because of their well-defined constraints. However, when data is not adequately constrained - like the online social networking scene, these constraints break down and learning algorithms face the challenge of adapting to indeterministic non-IID inputs for similar estimation accuracies. Kernel masks are folds which model these non-linear characteristics of non-IID datasets. In fact, the term "IID" has been loosely used to identify the breakdown of key symptotically symmetric and reversible qualities within a data sparsification model [169], [170]. It is adequate to presume that for each data distribution, its corresponding sparsification model is just as unique [184]. Oftentimes, data is distributed uniquely towards an specific application in question. For example, in twitter, messages are passed from one person to another or a group in a series of short and sparse texts whereas on facebook, similar information is condensed into dense paragraphs as posts and usually only transfered from one person to another, unless he or she belongs to a community in question [185].

The reason why kernels often adopt different masks is simply because there are just as many unique non-linear sparsification models which require linear algorithms to conform to before being effective at performing their predefined (learning) tasks [186], [82]. Example, in a typical non-linear binary

classification task, the objective is to learn a non-linear classifier f which maps a feature dataset $x_t : t = 1, 2, 3..., T$ from a data space R of d dimensions into subsets R_a and R_b, of which a and b are the binary classes whose labeled outputs are $y_t \in +1, -1$ respectively. For adequate classification to occur, a kernel mask profiling this linear feature to non-linear binary classifier mapping must be identified and defined as $f : R^d \rightarrow R$ [83]. A classification rule can then be built to extract the sign of the kernel mapping function $\hat{y}_t = sgn(f(x_t))$ where $\hat{y}_t$ is the predicted class label. Intuitively, the confidence of the classifier prediction is then measured by the magnitude of the kernel transformation $|f(x_t)|$ [175]. In an OAL setting, such a kernel mapping determines the regret function $R(T)$ of the classifier by calculating the hinge loss errors across the data manifold. This is given as:

$$R(T) = \sum_{t=1}^{T} l_t(f_t) - \inf_{f} \sum_{t=1}^{T} l_t(f) \tag{2.31}$$

As can be seen from the example above, a key capability of Online Kernel Learning (OKL) is to solve for linearly seperable tasks through the use of kernel mapping transforms. OKL is often the default 'best' interface with shallow ANN architectures like SVMs, HMMs, GMMs, AR(p) and KR(n). Some recent studies on AOL is given in Table 2.08.

Ref.	Methods	Contribution	Weakness
[174]	Online Batch Learning - Classifier Ensemble	• Classifier Ensemble based design for batch learning of continuous data streams. • Minimal Variance rule for constrained instance labelling from a continuous data stream. • Weight updating rule for adaptive latent topic drifts.	• Classifiers are built from labeled partition of data chunks after learning. • Bayes decision rule is used for choosing classifier within ensemble based on conditional probabilities of lowest variance between built ensembles and current ensembles in the data stream. • Choosing between the three proposed sampling in real time to dynamic nature of topic drifts is a challenging process.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[176]	Hybrid Learning Strategies	• Standard incremental learning framework for data streams with change detection. • Variable Uncertainty with time-varying thresholds. • Uncertainty labeling with randomization.	• Change detection technique maintains accuracy of classifiers as topic errors increase over time due to drifts. • A major drawback of this technique is that classifiers have to be continually retrained and suffers from diminishing feature reuse. • By shrinking time windows at changes, this strategy is poorly adapted at capturing label change features at instances of sudden topic drifts. • This strategy suffers from the drawback of being redundant during stable data streams and a hypersensitivity at fuctuating topic concept changes in real time. • The Normal kernel distribution used in their study centers towards the fixed locality of sampling at $\mu = 1$ and is unable to uncover dynamic latent drifting topics occuring at anywhere within the concept space. • Choosing the variance δ of the normal kernel mask is also a challenging problem. A larger δ lowers probabilitiy densities across a wider spread of the concept region while a smaller δ increases probability densities to a restricted spread within the concept space. • Larger variances which spread randomizations of threshold labelling over a larger concept space are invariant to spot latent topic changes (e.g. a seminar or conference). • Lower variances which aggressively restricts the threshold labelling randomizations to a very narrow concept locality exhibit an extremely sensitive temporal behavior (e.g. BBC world news feed, etc.) which can reflect as performance instability of labelling classifiers.
[83]	Online Passive Aggressive (PA) Learning	• Passive-Aggressive Active (PAA) algorithms adaptation for online settings. • Online Binary PAA classifier. • Online Multi-Class PAA classifier. • Online Cost-Sensitive PAA classifier	• Passive Aggressive Algorithms updates whenever the hindge loss function is non-zero in both situations of correctly classified and mis-classified labels. • PAA algorithms strategizes queries of labels using a randomization rule for detecting topic drifts. • Proposed method grows with unbounded complexity - especially with increasing temporial dynamic topic drifts over time. • PAA Approaches and algorithms suffer from convergence issues over time as label reuse diminishes.

Continued on next page

—Continued from previous page

Ref.	Methods	Contribution	Weakness
[175]	Online Kernel Based Learning	• Sparse Aggressive Learning (SPA) method used to establish a bounded output of support vectors. • Regret bounded optimizers for learning expectations.	• Sparse Passive Aggressive approach stochastically samples incoming training examples as support vectors at higher losses. • The trade-off between adequate bounding of support vector size kernels and maximizing learning accuracies of the classifier requires careful selections of the threshold parameters α and β. • By implementation of a bounding constraint, approach is ill-conditioned to detect latent topic drifts due to sampling inconsistancies.
[172]	Online Deep Learning - Shallow ANN	• Online DNN architecture. • Backpropagation hedging strategy - Hedge Backpropagation (HBP). • Hedge weight prediction from low level feature reuse.	• Hedging strategy adaptively reduces the stack depth of the architecture during backpropagation - from high hedge scores detected at their respective layers. • Requires constant re-training of model in new topic context instances due to adaptive effective changes to model structure.

Table 2.08. Survey of key developments in AOL

2.7.2 Taking It Deep

Deep learning has recently become a hallmark of artificial intelligence that has proven powerful methods of learning classifiers and making predictions [162]. The very basics of deep network designs are individual layers of neurons stacked on top of each other. Lower layer neurons are connected to upper layer neurons through weighted activations [163]. Such a design emulates the way human biological brains are wired to process highly complex information. Deep neural networks first originated from the designs of a Simple Boltzmann Machine (BM). A simple BM features state nodes in two planes. The lower plane contain states at the input while the upper plane contain states at a hidden or an output layer [169], [170]. Figure 2.12 shows how a Simple Boltzmann Machine is being structured. An initial design for this architecture included intra state dependencies within each layer. Hence, individual states contained within their respective layers are interconnected by weighted links. The main flaw of this design indicates that states within the layers of the BM are not independent. Therefore, mappings between the

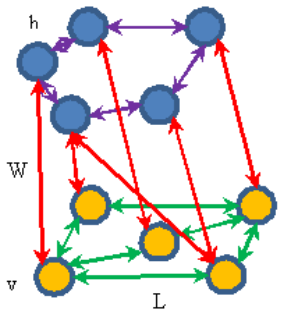


Figure 2.12. A Classical Boltzmann Machine - The top layer represents the stochastic binary hidden vector while the bottom layer denotes the stochastic binary visible variables.

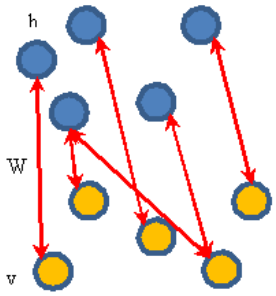


Figure 2.13. A Restricted Boltzmann Machine with no hidden to hidden and visible to visible intra layer connections

lower layer and the upper layer of the BM are not linear [76]. Computationally, learning the weights between just two layers alone of such a neural network architecture has proven to be extremely intensive. Later definitions of this model seek to simplify the relationships between the layers of the simple BM by imposing a constraint of independence between the states of each layer. This means that state nodes within each layer of the BM are restricted to contain connections from nodes only in between layers [138]. With the restriction of state independence within layers of the BM established, linear mappings between lower and upper layers can then be derived. This constraint greatly lifted the computational burden from learning neural network weights. Figure 2.13 shows a Restricted Boltzmann Machine (RBM) architecture.

Shallow Artificial Neural Networks

Simple BMs and RBMs respresent some of the earliest models of shallow learning framework to have implemented a neural network design. Architecturally, a simple BM is structurally defined as stochastically coupled binary unit pairs [138]. The coupling between visible (lower) layer $V \in \{0,1\}^D$ and hidden (upper) layer $H \in \{0,1\}^P$ vectors, expressed as: $\{V;H\}$ is driven by a characteristic Boltzmann energy state of interactivity as:

$$E(V,H,\theta) = -\frac{1}{2}V^T LV - \frac{1}{2}H^T JH - V^T WH \tag{2.32}$$

Where $\theta = \{W,J,L\}$ are Boltzmann Machine model weights between visible to hidden, visible to visible and hidden to hidden layers respectively. With the constraint of state node independence imposed on an RBM design, coupled intra-layered interactions between neurons are eliminated. This effectively reduces J (visible to visible) and L (hidden to hidden) weights to a fixed constant. The weight gradient at each epoch for ∇J and ∇L vanishes. This fundamentally rolls back the simple BM energy state equation to an RBM variant - defined as:

$$E(V,H,\theta) = -\frac{1}{2}V^T B - \frac{1}{2}H^T A - V^T WH \tag{2.33}$$

Where $B = LV$ denotes the visible layer state input bias and $A = JH$ denotes the hidden layer state activity bias.

Although BMs were the forerunners of shallow ANN architecture designs, their recursive generative stochastic architectures made it difficult to model multi-variate regressive states where neurons in hidden layers are directly and / or indirectly influenced by nodes from visible layers [137]. To address this problem, the perceptron was proposed. In the same vein, the perceptron is also a shallow ANN architecture. However, in difference, it is a discriminative feed-forward neural network [187]. A schematic design of the perceptron is given in Figure 2.14.

As with all DNN architectures designed for use in ANN learning techniques, it embodies three structurally critical core processes in the following order:

1. A Forward Pass (FP)
2. A subsequent Back Propagation (BP)

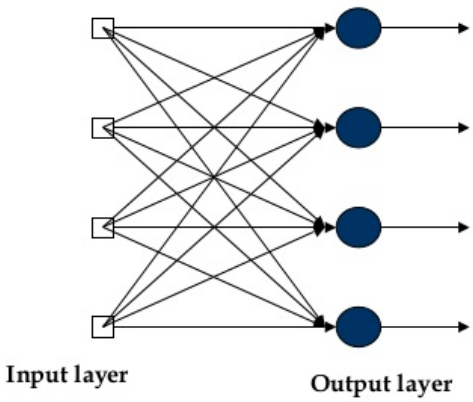


Figure 2.14. A feed forward single layer perceptron with input neurons directly connected to output neurons.

3. A final Weight Tuning (WT) phase

In addition, shallow ANN architectures contain the following key characteristics:

1. Instances are represented as many attribute-value pairs
2. Input values can be any logical or real values.
3. Target function output may be discrete-, real- or vector-valued.
4. Training examples may contain errors (Which is fine as long as error gradients vanish at convergence to the expectation).
5. Fast evaluation of the learned target function may be required.
6. Many iterations may be necessary to converge to a good approximation.
7. Ability of humans to understand the learned target function is unimportant.
8. Learned weights do not have to be intuitively understandable (only requirement - machine interpretable).

Discriminative Multi-Layered Frameworks

A perceptron represents a hyperplane of decision surfaces in an n-dimensional space of instances [121]. In this space, some data example sets are linearly separable while others cannot be extended apart by similar boundaries. Functionally, perceptrons are also capable of representing many boolean functions like the AND, NAND, NOR and OR [187]. In essence, the perceptron neuron takes in a vector of real valued inputs $\vec{X}$ and its corresponding weights $\vec{W}$

and proceeds to calculate their linear combination to produce and output:

$$Y_i = \sum_{i=0}^{n} W_i X_i \tag{2.34}$$

This occurs for as many neurons as there are in a single perceptron layer. Activations of neurons in an ANN architecture is often achieved through an activation function. One of the earliest adopted activation functions for such neurons is the Sigmoid function (also known as the logistic squashing function) given as:

$$\sigma = \frac{1}{1 + exp^{-x}} \tag{2.35}$$

There are other variants of neuron activation functions (e.g. tanh, ReLU, ELU, Softmax, step, etc.) which also achieve the same effects of firing the neuron with an output if a weighted threshold w_0 is exceeded [163]. The objective function of ANN architectures (like the perceptron) is to minimize prediction errors at the output. This means $\underset{\to 0}{min}(E_t = E(\vec{X})_t - Y_t)$ where E_t is the error at the t-epoch, $E(\vec{X})_t$ is the expectation from input $\vec{X}$ and Y_t is the predicted output. A training update rule optimizes this objective function by minimizing this error as:

$$w_i = w_i + \triangle w_i \tag{2.36}$$

Where $\triangle w_i = \eta(E(\vec{X})_t - Y_t)x_i$. Here, η is specified as the learning rate. The learning rate is a tunable parameter which determines how quickly the algorithm converges to an estimation of the output. A drawback to this is that the learning rate has to be trepidatiously selected for the learning algorithm to benefit from both performance (fast convergence) and accuracy (error minimization) [162]. If η is too high, then the ANN converges quickly, but with large prediction errors. However, if η is too low, then ANNs converges slowly, nonetheless with high prediction accuracies.

Shallow ANN architectures like the Recurrent Neural Networks (RNN) in Figure 2.15, Long Short Term Memory architectures (LSTMs) in Figure 2.16 and Extreme Learning Machines (ELMs) Figure 2.17. are versatile enough to learn relations between features of a small dimensional problem. However, as network complexity grows, increasingly larger arrays of neurons are required

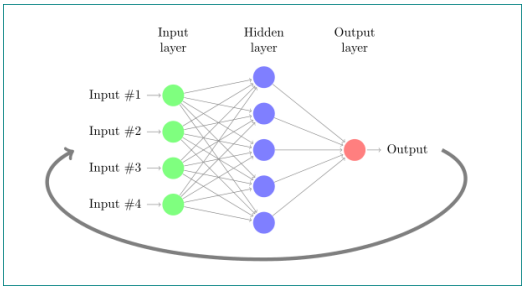


Figure 2.15. A Recurrent Neural Network where outputs are fed back as input to predict the next hidden confabulation layer to produce the next output in a predefined sequence.

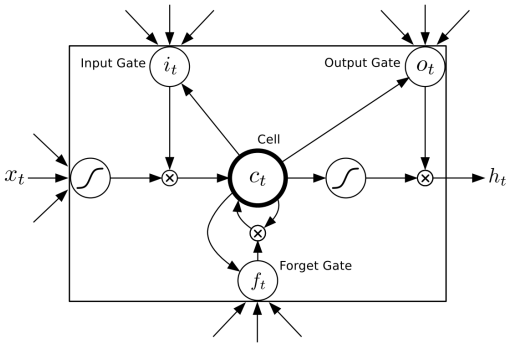


Figure 2.16. An LSTM architecture with recurrent inputs and gates that 'forgets' non-relevant knowledge and cells that retain memory over useful knowledge of its current task.

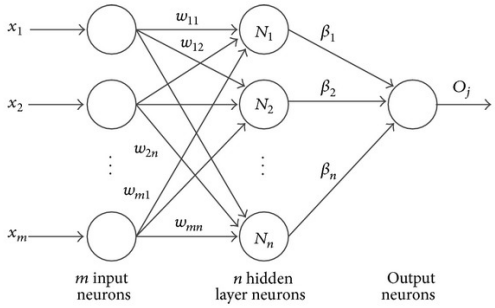


Figure 2.17. An ELM architecture which contains a single hidden layer that provides non-linear transformations between input and output neurons.

to represent high convolutions of information in real-life problem scales. Linear representative transformations of single layered ANNs break down. In order to maintain simplicity of neuron representations per layer and linearity of mapping in transformations between layers, Multilayer ANNs like the Multi-Layer Perceptron (MLP) in Figure 2.18 have been proposed.

Essentially, Multilayer ANNs represent a schema of stacking individual

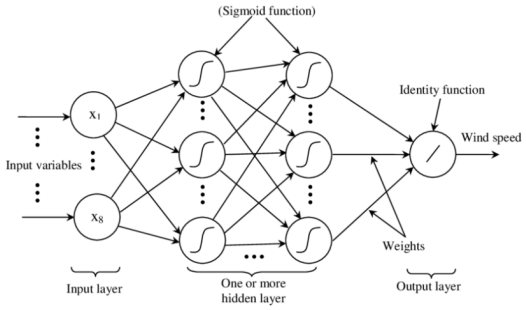


Figure 2.18. An MLP architecture where input neurons and sequentially transformed from visible through hidden confabulations to finally arrive at the obeserved output layers.

layers on top of one another. In this architecture, the output of a single layer becomes the input to the layer above it. Such a design enables key features of distinct ANN structures (like discriminative feed-forward perceptrons and loop-back RNNs) to be preserved. More specifically, Multilayer ANNs maintain the following key characteristics:

1. Capable of learning nonlinear decision surfaces.
2. Normally directed and acyclic - Feed Forward Network.
3. Based on a squashed activation function.
4. Extends linear learning complexity to a higher dimensional problem set from a single layer to multiple layers.
5. Is able to represent a wide range of functions (e.g. Boolean functions as single layer, Continuous functions as double layers, arbitrary functions as three layers, etc.).
6. Terminates when error threshold condition is met.

A summary of developed shallow ANN architectures is given in Table 2.09.

Ref.	Methods	Contribution	Weakness
[188]	Multi-Task Learning (MTL)	• Improved Multi-Task Learning based on non-convex alternating structure optimization (iASO) algorithm. • ASO non-convex to convex formulation conversion. • Scalable Alternating Optimization (cASO) for large datasets.	• Learns linear predictions based on structure optimization within a shared feature space. • Exploits a low-dimensional (1-2 layers) map across m number of tasks. • Search manifold confined to feasible sets of a convex region. • Unable to scale to high complexities and concave solutions.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[36]	Hidden Markov Models (HMM)	• Event prediction classification problem based on Bayes Decision. • GDELT (Global Data on Events, Location and Tone) formulation of HMMs. • Prediction model of country stability.	• Shallow architecture inadequate to represent complex signals like country stability. • Prediction performance converges fast due to shallow architectures, however, at the cost of accuracy in ground truth estimations.
[137]	Parallel Tampering (PT) - RBM	• Improved Markov Chain Monte Carlo (MCMC) sampling technique. • Parallel Tampering approach to sampling distributions for learning RBMs. • Alternative method of state evalution at every learning update step.	• Method scales poorly to large data complexity. • Performance of model is highly sensitive to manually tuned parameters. • Requires re-training for every new data instance.
[189]	Conditional Random Fields (CRFs)	• Implements CRF learning design to model retweet patterns. • Includes three user-tweet feature types. • Graph partition and Network relation construction for retweet prediction.	• Constrains the 'tweet' solution space to a small world model using graph partitioning methods. • Modular solutions of small world twitter spaces using conditional random fields fails to address problems associated with topic drifts. • Static similarity feature based approach is adequate enough for CRF models. However, dynamic feature metrics still remain a challenge to represent with shallow ANN architectures.
[190]	Logistic Regression (LR) - KNN	• Feature filter to test against associated outcome score of relevancy and significance. • Wrapper algorithms for optimal classifier selection. • Principle Component Analysis (PCA) for data dimensionality reduction. • Performance evaluations of learning classifiers: k-nearest neighbors, logistic regresion and Cox regression.	• Model scales poorly to large data dimensionalities. • Model is ill-positioned to represent high data sophistication of patient data.
[81]	Maximum Entropy (maxEnt)	• Topic Maximum Entropy (TME) model for social emotion classification. • Latent topic modeling. • Transformation mapping of features onto concept space. • Emotion classification over data sparsity.	• A major drawback is that if model is used in longer texts with higher sentimental complexities, topic drift detection fails and predictions become inaccurate. • maxEnt scales well to well-defined and self-contained short texts. However, lacks the representational power for larget more complex datasets. • TME is restricted to batch learning of N-gram words and cannot be used for continuous sentiment stream.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[191]	Restricted Boltzmann Machine (RBM)	• Similarity based Neural Language Model (NLM) for mapping medical word representations (Skip-Gram). • Medical Concept to Terms mapping. • Improves semantic similarity measure for medical information retrieval and informatics.	• Computation of output words to medical concepts are very intensive for shallow ANN learning architectures. • Similarity based Skip-Gram NLM is restricted to batch learning over a sliding concept sequence window radius. This shallow ANN architecure will run into representational issues when faced with a larger corpus of contrustive medical terms and concepts.
[192]	Recursive Neural Networks (RNN-LSTM)	• Nested LSTM architecture. • Multi-layer memory strategy. • Memory feature concatenation of inputs.	• Nested LSTM structure stores feature memories hierarchically in a manner similar to Stacked LSTMs. • However, Nested LSTMs have access to inner memories which are more efficient temporal abstractions of the entire memorized datasets.
[138]	Multi-Layered Restricted Boltzmann Machine (RBM)	• Mesh Convolutional Restricted Boltzmann Machine (MCRBM). • Novel local energy function as model data input. • Mesh Convolutional Deep Belief Networks (MCDBN).	• Both visible and hidden layer neurons are irregularly organized on a 3-D mesh surface. • Deviation from traditional architectures require irregular input and output vector structure. • Computationally intensive through the use of local energy function distributions as training input.
[88]	Autoencoder	• Multi-layer encoder and decoder network. • Stacked RBM architecture.	• Requires discovery of good initial weights for training of multiple hidden layers. • Architecture is restricted to shallow ANN types because of early difficulties with vanishing gradients.

Continued on next page

—Continued from previous page

Ref.	Methods	Contribution	Weakness
[193]	1-hot Deep Stacking Networks (DSN)	• Single Layer Deep Stacking Network (DSN) - MLP as the authors call it. • Med2Vec architecture. • 3-layer Stacked Autoencoder.	• Med2Vec learns code to visit representations by concatanating visit and demographic vectors as inputs into the stack. • Minimizing cross entropy losses yields squared $[O(T\bar{M}^2\|C\|m)]$ and linear $O(Tw(\|C\|(m+n)+mn))$ complexities for code-level and visit-level objectives . • Overall objective function complexity is a squared relation of medical codes as $O(T\|C\|m(\bar{M}^2+w))$. • Med2Vec is unstable during the convergence process which may lead to a loss of generality and vanishing gradients if the depth of the neural network increases to represent more complex data sets. • From computational complexity of optimizing objective functions, Med2Vec takes the longest time to train for a given data set of visits as compared to traditional learning methods like Stacked Autoencoder and GloVE+. • Prediction performance of Med2Vec closely correlates to Stacked Autoencoder architectures.

Table 2.09. Table of key developments in Shallow ANNs

2.7.3 Taking It Deeper

Deep Neural Networks (DNNs) can be categorized into three general architectures: Generative, Discriminative and Hybrid. Each one of these forms achieves a different and distinct learning objective function to the other [194], [162]. Intrinsically, they have a similar layered architecture where classification and information processing occurs in stages from input to output neuron signals. A deep architecture generally consists of more than 10 layered neurons in a stack which sequentially processes information for applications like pattern classifications and feature representations. They differ from shallow learning architectures in two distinct aspects: rate of convergence and complexity of learning tasks [163]. Examples of some shallow learning architectures include:

- Gaussian Mixture Models (GMMs)
- Hidden Markov Models (HMMs)

- Conditional Random Fields (CRFs)
- Maximum Entropy (MaxEnt)
- Support Vector Machines (SVM)
- Logistic Regression (AR(p), VAR)
- Kernel Regression (KR(n), KVA)
- Multi-Layered Perceptron (MLP)

Shallow architectures guarantee a faster rate of learning convergence. However, this is often done at the prime cost of scaling back on complexity. A common characteristic of shallow architectures is that they consist mainly of a single transformational hyper-space where inputs are mapped onto an output that is oftentimes unobservable [116]. For example, SVM traditionally relies on a linear function to determine optimal euclidean data feature seperations. This transformation takes place from the training set onto a classifier space. For non-linear mappings, SVM requires the use of a kernel mask in order to maintain this 1-step transitional approach. Without which, its classifiers are unable to adapt to this non-linear sophistication in the observables [163], [162]. In contrast however, deep architectures contain stacked layers on top of one another that chain the transformations of interpreting the input hyperspace signals to an output expectation. For example, perception applications where machines are expected perform a wide varietly of tasks (like segmentation, detection, identification, tracking, optical flow, recognition, etc.) from analyzing image frames, are synonomous to hierarchical learning models. The visible layer is oftentimes made up of complex signal structures that represent a rich but convoluted hyper-set of image metadata. Although easily interpretable by biological cognative processes, they remain a challenge to machines and their limited computational power [195]. A deep architecture allows these 'signal convolutions' to be deconvolved into simpler fundamental forms and captured within each layer. Each of these layers contain embedded informational intelligence which are dependent on the 'truth' contributions from their corresponding layers below [168]. As a result, while shallow ANN architectures have shown to be efficient in solving simple and well-constrainted problems, they lack the representational power and robust modeling to deal with highly sophisticated real world signals. In contrast, DNN architectures model nature's hierarchy of intelligence by establishing multiple layers of non-linear (complex) processing stages where lower layers contribute to higher layers of information 'truths' which in turn, then

convolves to become an observable sophisticated output signal.

Generative DNN

Generative DNNs are a class of neural networks which have evolved into deep architectures from shallow stochastic relational learning models. Their objective functions are designed to characterize joint statistical distributions of visible data and their associated classes. This is often achieved through cross-validations between observables and expectations by extracting high-order correlations between input data and its corresponding generative posteriors.

Generative DNNs apply the concept of "unsupervised pre-training" as a primary stochastic approach to establish activity at each layer. Starting at the input (lowest) layer of a DNN, generative models sequentially learn expectations of neuron state distributions in a layer-by-layer manner as it progresses upwards. This is often done without inferring information from upper layers to bias its expectations. Greedy algorithms together with sampling mechanisms (e.g. gibbs sampler) are often used to sample from the largest possible manifold of posterior state distributions. As a result, this type of DNNs is highly adaptable and versatile in expressing very complex correlations between realistic feature observables to data inputs through stochastic estimations. An added strength of this model is that the generative processes procedurally addresses some aspects of overfitting and underfitting - which are usually big drawbacks of DNN models. This is achieved through selective sampling of distributions where only the highest contributing features are filtered into individual layer designs. Additionally, generative DNNs also scale well to the data sparsity, ergodicity and asymptotic convergence. Given large enough unsupervised iterative runs, generative DNNs are capable of building highly accurate estimations of unknown ground truth distributions. Prominant applications of such processes include Natural Language Processing (NLP), Image to word (e.g. Hashtag) mapping, Speech to language to word translation, etc. However, these architectures trade off representational efficacies at the prime cost of speed. Not only do individual activity layers of the DNN architecure have to be determined. The adequate propagation depth of the entire DNN model - corresponding to the sophistication of the

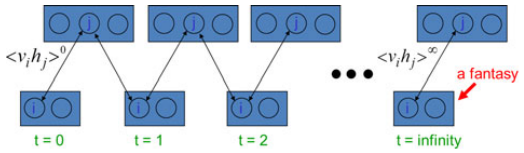


Figure 2.19. A Generative RNN where posterior distortions are generated recursively from output to input feedbacks

observable needs to be derived as well.

Prominant generative DNNs include deep autoencoders (ADNN), deep boltzmann machines (DBM) and deep belief networks (DBM). Although RNNs have been considered shallow architectures because of their single layer designs, they can be regarded as a special class of generative DNNs when used to model and generate sequantial data. The depth of the model formed from these iterations when coupled with successive training vectors can be as large as the length of the input data sequence. However, they have not been as popular as other DNN models because of vanishing gradients. Figure 2.19 shows the architecture of a generative RNN.

Discriminative DNN

By the application of Bayes rule $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ - which states simply that the probability of an event A occuring is conditionally dependent on some apriori knowledge of an event B occuring before A. A generative DNN is empowered to provide discriminative inferences through the sampling process of its generated posterior states from the input. This is often done through characterizations of the sampled posterior distributions to the target classifiers of the expectations. A wide variety of shallow architectures such as HMMs and CRFs objectively functions discriminatively. For example, CRFs can be considered as a class of shallow ANNs which statistically predicts labels under a well-defined and fully structured environment. A discriminative function of single hidden layer CRFs is given as:

$$P(Y|X) = \sum_{H} P(y|h,x)P(h|x) \tag{2.37}$$

Where $h \in H$ is the neuron activity within the hidden layer H. Conditionally, it states that the prediction from a CRF given a certain set of inputs is conditionally discriminative against an observed expectation.

Similarly, HMMs can be objectively expressed as:

$$P(Y) = \sum_{h=0,l=0}^{(L-1)} P(Y_l|X_h)P(X_h) \tag{2.38}$$

Where Y_l are individual visible observations at the output and X_h are individual states across the hidden markov layer.

Deep discriminative structures are easily formed from stacking outputs of shallow ANN architectures into inputs of similar shallow ANN structures above. A good example of such networks formed from this stacking process of outputs to inputs is known as the Deep Stacking Network (DSN). In a DSN architecture, the output of one layer concatenates with additional sequential features to form a new set of inputs into the next layer. Thus, a DSN layer discriminative output prediction is only conditionally dependent on truth distributions from lower layers. Figure 2.20 shows the structural overview of a DSN.

The characteristic output equation of the network is simply:

$$y_n = U^T h_n = U^T \sigma(W^T x_n) = G_n(U, W) \tag{2.39}$$

Where y_n is the n-th epoch output vector of the network, U^T is the upper layer weight vector, h_n is the n-th epoch hidden layer activity vector.

Hybrid DNN

Hybrid DNNs couple unique features from both generative and discriminative models to provide better learning optimization and regularization. The general objective function of Hybrid DNNs is still discriminative classification over data observables. However, such labelling activity is oftentimes heavily assisted by characterizations of joint likelihood distributions extracted from

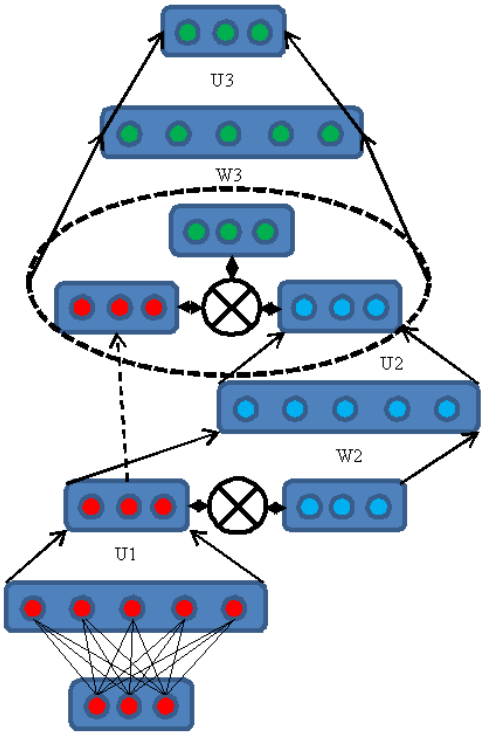


Figure 2.20. A DSN architecture where output distortions of individual successive layers in the deep network stack are convolved together with progressive training signal inputs from external sources.

generative structural counterparts. Although hybrid DNNs retain aspects of their parent super-architectures, they need not necessarily be probabilistic (e.g. hybrid deep autoencoders). While DNNs are more efficient to train and adequate for learning complex systems - a characteristic which is attributed to its loosely constrained model construction. Deep probabilistic models (like Deep Bayesian Network (DBN) and Deep Boltzmann Machines (DBM)) are more adapt at representational and interpretational features of latent domain knowledge. Their architectural design compromises learning and inference mechanisms for improved accuracies at the handling of relational uncertainties.

One of the inherited strengths of hybrid models are that they are capable of providing an excellent set of initialization points in a highly complex, non-linear estimation environment. Additionally, they are also able to maintain adequate representational control over overall complexities of the entire model. Appications like speech to text and language translations have ben-

efited from both generative and disriminative attributes of hybrid models.

Overview of Deep Structures

A summary of related developments using deep learning architectures is given in Table 2.10.

Ref.	Methods	Contribution	Weakness
[196]	Hybrid DNN - TDSN	• Tensorized Deep Stacking Network (TDSN). • Tensor bilinear predictions from two parallel hidden layers within a single DNN architecture. • Extends DSN architectures to include multi-modal tensorized predictions in different dimensions.	• TDSN architectures are highly versitle and capable of handling natural phonetic information in high ordered complex dimensions. • Tensorized DNN architectures addresses limitations in scalability and parallelization of learning algorithms for DNNs. • Comparable performance to traditional DNN architectures. • Eliminates need for hard scaled, sequential fine-tuning. • Computationally intensive. Requires the use of parallel processing (GPGPUs) to converge within a reasonable amount of time frame.
[197]	Discriminative DNN - CNN	• Extended CNN model which strengthens position sensitivity. • Incoporates history awareness into CNN model for sequence data transformations. • Asymmetric positional information interpretation by the novel PoseNet architecture.	• CNN discriminative architectures highly capable of representing complex information but lacks time-space awareness. • PoseNet architecture implements a context-sensitive awareness to improve sequence to sequence learning accuracy. • Sequence to sequence learning in CNN is capable of handling complex problems in a multi-level stack of representations. • Sequential learning in PoseNet accures a fractional cost function of $O(n/k)$, where n is the number of input elements and k is the kernel batch size. • PoseNet architecture is computationally efficient. However, unable to scale to non-linearities of input data against observables sequentially. • PoseNet architecture is relatively shallow with only 6 encoding and 5 decoding layers.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[198]	Discriminative DNN - DsN	• Deeply Supervised Network (DsN) architecture which extends over limitations of traditional CNN learning structures. • DsN architecture integrates a "champion" hidden layer objective function for streamlining upper layer input selection process for minimising target loss.	• DsN strategizes learning process by providing integrated direct supervision at individual hidden layers. • Softens hidden layer constraints into new regularizations as champion objectives for effective supervised training at hidden layers. • Benefits of DsN architectures include strong "regularizations" for shallow ANN architectures and significant performance gains for complex DNN structures. • The main drawback is that the network has to be optimized at each hidden layer according to individual champion objectives. If the stack is large (i.e. a deep architecture), then learning output objectives converges very slowly. • Supervised training of hidden layers require human interpretation of the feature states at each iterative forward step. This is oftentimes unknown and hard to visualize.
[199]	Discriminative DNN - Residual Networks	• Skip connections DNN architecture. • Novel pre-training procedure. • Improved training method to residual networks.	• Identity mapping pre-training stragegy allows short networks to be trained at pre-training phases and deep networks to be used at run time. • Direct benefits are a reduction in training time and an improvement in feature reuse and avoidance of vanishing gradients. • Integrates the random removal of a substantial fraction of layers at once through identity transformations to reduce stack size for pre-training. • Although approximations to ground truths of observables at the output may still be preserved (as evidenced in their study results), this approach will result in poor estimations due to wide variances of large input datasets. • Identity function mappings of Residual Networks (ResNets) from a lower layer to an upper layer block jump may inaccurately represent hidden layer activations from one feature state layer to the next.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[200]	multi-dimensional Hybrid DNNs - MultiTask Reinforcement Learning	• Importance Weighted Actor-Learner Architecture (IMPALA). • V-trace off policy actor-critic algorithm. • Diverse cognitive tasking environment.	• IMPALA architectures are extremely scalable to large numbers of learning machines without sacrificing training stability or data efficiency. • Aggressive parallelization of learning tasks from mini-batch trajectory updates. • V-trace target algorithm improves update lag between policy and learner by several steps. • Learning trajectories are generally deterministic - which does not provide for very good scalability to generalization of tasks. • Computationally intensive. Requires the use of parallel processing (GPGPUs) to converge multi-tasking opertaions within a reasonable amount of time frame.
[201]	Hybrid DNN - non-linear parameter estimation	• Stable forward propagation. • Asymmetric Weight Matrix casting forward propagation. • Discretized Hamiltonian neural network energy masks. • Hamiltonian continuous form symplectic integration forward propagation.	• Stable Forward Propagation (FP) architectures normalizes sensitivity of output observables to input signal disturbances. • Under stable FP normalization startegies, exploding and vanishing gradients can be effectively controlled by adding discrete transformational states as manifold constraints to optimize the objective function adequately. • FP transform constraints diminish the adaptability of the DNN model by restricting the manifold characterizations to data feature specific training processes in question. • In exchange, these constraints are able to directly address ill-posed learning stability problems of most DNN architectures like vanishing and exploding gradients from a model generalization perspective. • Derivative based regularization derives small time steps for well-posedness of a learning task. This could lead to multiple iterations and long convergence times to learning an objective.

Continued on next page

—*Continued from previous page*

Ref.	Methods	Contribution	Weakness
[202]	Discriminative DNNs - FHIR	• Generic data processing pipeline for Fast Healthcare Interoperability Resources (FHIR). • Demonstrate effectiveness of deep learning models to a wide variety of predictive problems and settings in hospital healthcare records.	• Data processing phase accepts raw EHR data as inputs and produces FHIR outputs without patient record harmonization. • Unified data structure for generalized deep model prediction tasks. • Predicted outcomes from diverse domains include: Inpatient mortality, 30-day planned readmission and Long length of stay. • Final prediction across hospital datasets relies on an ensemble model made up of RNN-LSTM, Time-Aware TANN and Time-boosted decision stumps. • DNN Models developed for this study is still relatively shallow and cannot represent data complexities from all 3 datasets adequately in one single deep learning structure.
[139]	Hybrid DNN - CNN variants	• Convolutional Neural Network based extreme multi-label text classification deep learning architecture (XML-CNN). • Dynamic Max Pooling to capture a richer set of information. • Hidden bottleneck layer within architecture between pooling and output layers of the CNN structure.	• XML-CNN combines representational strengths of DNNs and multi-label text classifiers to improve performance of existing XMTC approaches. • Loss of generalization due to unstable convergence behaviors. • Requires relatively longer training times as compared to other learning models.
[203]	Dense Document to Vector TFIDF	• D2V and Classical TFIDF training input vector concatenation. • Binary classifier. • Recommendation system of the LTR framework.	• DeepMeSH generates a dense D2V semantic vector representation of each document framework. • TF-IDF vector is derived to compute document to medical topic alignments. • Relevence scores from prediction integrating dense D2V representations and TF-IDF calculations provide a ranking criterion of individual documents. • Subject Heading labeller integrates the MeSHRanker and MeSHNumber predictions. • Results show that D2V TF-IDF classifiers achieved the highest F-measure score of 0.6033.

Table 2.10. Table of key developments in DNNs

2.7.4 Fractal Intelligence

Deep learning and their associated neural network structure was first proposed for open research in [88]. With the advance of technology (CPUs and

GPGPUs), computational power has overtaken lengthy training processes of the 4 layer autoencoder proposed in their study. Training the architecture made up of stacked RBMs today, now converges to a good approximation within a few seconds. However, there are still drawbacks to using DNNs and a vast array of ML techniques in today's contextual world. One of the main disadvantages is the fact that DNNs have to be re-trained for specific tasks which they seek to solve. The function trained within one DNN model is hard-constrained to that specific task which it has been objectified to learn a policy for. A good set of visible layer weights which act as optimal initialization vectors for the DNN model which has been learned for that task may prove abysmal in another task setting with a different DNN model. In deep multi-layered learning architectures, common problems which back propagation runs into are vanishing and exploding gradients due to poor initialization of weights. Fine-tuning and randomization of low level weights destroys previously learned knowledge of objective functions. Generalizing DNNs and their structures across a variety of tasks without successive pre-training steps requires a transfer learning function which remembers the neuron activity mappings across all tasks. In addition, it also requires a similarity measure between tasks to determine how much latent information within a set of previously initialized weights needs to be retained for subsequent tasks. [204] approaches this problem by distilling fine grained classes from a cumbersome model (e.g. a very deep neural net or an ensemble) to a smaller more agile architectures for deployment. A key development to this research involves picking out higher denominations (HD) of redundant knowledge learned across the ensemble and keeping only the lowest common denominations (LCD) of training that are relevent across all models in the ensemble. Doing so facilitates transfer learning into a single smaller model of choice without catastrophic forgetting. In situational aspects of learning applications like OAL where models are expected to converge relatively quickly within limited amounts of given resources (e.g. time and space) to an expectation of the observable, knowledge distillation from high representational models like DNNs transfered into lithe and minature models like PAA offers an attractive solution.

Knowledge Distillation

In comparison to knowledge of complex features learned from ultra-deep models (1000 over layers), distillation is a staged process that successively compresses information into iteratively higher reduced forms using the temperature spread of a softmax kernel [204]. From a highly descriptive knowledge vector of learned structures, information is reducabily fractioned and compressed at each stage till it reaches an irreducible satisfiability. Beyond which, further distillation requires convergence towards a target objective function (e.g. similarity descriptor between a specific sub-set of tasks between learned models in an ensemble) [204]. This is because information loss during data reduction at this stage is non-reversible. The LCD residual knowledge retained from this process are also known as structural fractals.

Structural Fractals

Fractal networks are repeated truncated sub-structures of an elaborately architectured DNN [26], [18], [29]. They are a subset of AI techniques which are used within the domains of fractal intelligence in relationships of OSNs to describe structures of chaos [29]. More importantly, these sub-structures retain latent knowledge learned from tasks defined by its parent super-structures [18]. More interestingly as well, is their key ability to transition between generating deeply stacked DNNs for powerful representations of sophisticated information or shrinking the cumbersome architectures down into a shallow design with just a few repeating layers for training and OAL deployments [26], [18]. The crucial task however, is in the detection and identification of the repeating "structural fractal" patterns in a given DNN architecture design. For example, 'anytime' fractal designs of shallow to deep NNs addresses problems of selecting adequate network depths in [172]. Although the authors in their work have used a hedging strategy to determine (backpropagation) network depth based on high performing hedged classifiers at each layer of the structural design for shortened stacks of backpropagation. A huge drawback to their approach is that hedging backpropagation still requires the constant updates of weights as ground truths are continuously revealed in a data stream. This not only means that, learning hedge param-

eters during online gradient descent necessitates additional learning costs (in terms of time and computational resources) - which can easily scale up in complexity to actual DNN model depth sizes for highly sophisticated feature representations. Most importantly, it means also that the authors have overlooked the sensitivity aspects of their hedging approach towards fluctuating latent concept drifts. If the residual effects of these updates are not adequately controlled during learning phases of the hedging parameters, their online deep neural network structures will 'bounce' back and forth in a springing motion between adjusted and re-adjusted backprop depths of a pre-defined architecture. This not only causes instability in convergence but also generates ill-posed visible layer initialization weights for every learning / training iteration. As a result, transfer learning from shallow to deep architectures are effectively redundant as the network needs to re-learn all weights again based on another given instance of an identified topic. Additionally, vanishing-exploding gradients do not entirely disappear - especially when upper layers closer to the output have been hedged in for back propagation for instantaneuously complex topic drift representations. Furthermore, hedge weighted predictions are not always accurate ground truth expectations. To circumvent these problems, a fractal network strategy which distills knowledge learnt from deep networks at every new instance and transfers it into shallow ANN architectures (that are computationally agile enough to perform tasks like real-time prediction and classification in an online setting) is the best solution forward to tackling long-standing problems of continual or lifelong learning.

Key Developments On Fractal Intelligence

A summary of key developments and studies on fractal intelligence are revealed in Table 2.11.

Ref.	Methods	Contribution	Weakness
[204]	Knowledge Distillation	• Information compression using distillation technique. • Optimizes transfer learning from cumbersome ensemble model into a single model.	• Distillation technique uses the temperature spread of a softmax mask over target distributions in a transfer set. • Setting the adequate temperature for optimal spread during distillation process can be a challenge. • Mis-classified labels will propagate through the distillation process if spread is too narrow (i.e. softmax function too rigid). • Experiments on MNIST and Automatic Speech Recognition (ASR) show significant improvements of the distlled learning model from target ensemble sets to a similar sized model that is directly learned from the same training data. • Higher distillation temperature softmax profiles require wider allocation of information bandwidth for knowledge transfer. • Determining information bandwidth required for different softmax temperature profiles for a target ensemble is a challenging and cannot be efficiently determined. • Insufficient allocation of information bandwidth could result in knowledge corruption at the output receiving end (the distilled model) of the distillation process.
[26]	Fractal Networks	• Communication Kernel Skeleton Architecture. • Fractal Scaling power-law relation of network super-structure reconstruction.	• Although fractal networks contain similar properties to its parent structure, it does not inherit enough descriptors to rebuild the original parent network in an adequate manner. • Determining skeletal paths from a spanning tree is often a challenging process that requires modularity in the sparsification of the dataset. • Degree distributions are also not always preserved between fine-grained and course-grain perspectives of the parent network.
[18]	Fractal Networks	• Fractal model of repeated truncated networks. • FracalNet - an alternative extension of ResNet. • Connection euclideations between FractalNet and traditional DNN designs. • Drop-path regularization protocol.	• Computationally intensive. Requires the use of parallel processing (GPGPUs) to discover new fractals within a DNN structure generated for a new task instance. • Trades off fractal subnetwork discovery complexities with efficient and short training times of fractals in which partial evaluation yields good initialization weights for larger superstructures of traditional DNNs.

Table 2.11. Table of key developments in Fractal Intelligence

2.8 Performance and Evaluation

Well known measurement indicators can be used for testing accuracy, performance and reliability of model data from the recognition process. These include:

1. The pearson-r correlation which follows the mathematical model:

$$r = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}} \tag{2.40}$$

   Where x and y belong to distributions of two bivariate datasets to be tested. Essentially, the relational coefficient r, measures the ratio of the standard deviations between the two values against their variances. This test can be used on wide ranging results acquired from the mid to high level interpretations (identification and classification) of the model design to determine the correlation ranking measure between expectation and ground truths.

2. The Pearson Kendall correlation which follows the mathematical model:

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{2.41}$$

   Where for a data size n of two bags of data values, n_c and n_d refer to the number of concordant and discordant pairings. This test can be used across multiple phases of the recognition model process to measure the strength of association between relational intelligence and their evolutionary structural patterns.

3. The Spearman correlation which follows the relation:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{2.42}$$

   Where ρ is the spearman rank correlation, d_i is the rank difference between the experimental datasets of study and n is the number of observations. This test is also used on the results in the second phase of the recognition model design to measure the ranked monoticity between relational profiles and structural patterns of the online social network.

4. The k-fold cross-validation test to determine the proficiency of selected OAI models in an ensemble machine for the handling of recognition task

objectives. This method is used at the very last phase to establish the reilability, accuracy and utility of the proposed relational intelligence recognition model against ground truth occurances.

5. The AUC-ROC integral performance scores which follows the mathematical relation:

$$AUC = \frac{1}{PN} \sum_{i=1}^{P} \sum_{j=1}^{N} \psi_{ij} \tag{2.43}$$

Where P is the number of positives, N is the number of negatives and ψ is the data pair of both baseline and calculated scores. This test is suitable for use in extended phases of the recognition process to determine accuracy performance of the final predictions.

2.9 Conclusion And Trending Directions

There has been explosive growth in structural techniques of analysing complex heterogeneous information networks across several forms of representation. However, there is little study and focus on the affective and sentimenal recognition of evolving relational patterns in a wide range of practical applications. This chapter provides an extensive survey of this young and emerging field. Recent developments of data representation techniques and their relevence on structural integrity along with future research directions are presented. Although the framework presented in this study apprears roughly conditioned for recognition tasks, a lot more work on Evolutionary Artificial Intelligence (EAI) needs to be done. Furthermore, existing fundamental issues on structural integrity in a given information topology needs to be thoroughly understood for adequate facilitation of the recognition task. The future trend of social networking computing is improving and progressing through the years and humans could see new methods of knowledge and information discovery being developed to intelligently gather data.

Objectively, this chapter has provided an extentive overview to state-of-the-art methods and approaches which were both researched and implemented for recognizing key relational intelligent behaviours in Online Social Networks (OSN). The main motivation of presenting this chapter is to provide a solid foundational background of preliminary knowledge to the techniques

and mechanisms used in analyzing relational features of OSNs; which can then be adapted to represent relational flux and turbulence in active online social transactions. Although developments to mainstream tasks like community detection, link prediction, node identification, etc. have been vast in recent study. To the best knowledge of existing literature however, there has not been a survey which examines the problem of computationally describing and recognizing relational states and communicative behaviors through OSNs. This chapter provides a thorough introduction to current AI modalities which attempt to solve this problem of complexity and importantly, leads on to further research developments in the subsequent chapters of this thesis - where the highly sophisticated task of predicting generalized events is adequately addressed from a relational intelligence perspective.

CHAPTER 3

PREDICTING LINK STABILITY IN OSN

3.1 Introduction

The Internet today is a platform where social transactions are staged by the billions per second all around the globe. Although it has brought several benefits by its technological advancements, it also opens up many other problems which are largely relationally focused. Some of which include but are not limited to: Exponentially increasing data privacy intrusions on a yearly trend [31], rising number of internet suicides from online depression [205], [31], Account poisioning and hacking [206], [31], Terrorism and security breaches [206], [31], Information warfare and cyber attacks [31].

From a structural viewpoint, popular networks like Google, Facebook, Twitter, Youtube, etc. are often used as social and affective means to express exchanges and status of evolving human ties [206]. This is often done through rich expanses of emotional and sentimental fidelities which fluctuate over topic drifts [206]. Stable links are defined as relations (both positive and negative) where emotional flux remains relatively high through social evolution [207], [31].

Detecting stable links within online social networks is important in many real-life applications. For example, stable links can specifically be applied to analyze and solve interesting problems like detecting a disease outbreak within a community, controlling privacy in networks, detecting fraud and outliers, identifying spam in emails, etc [208]. Identifying stable relations within a social circle as structural pillars of a community is also very important in abating cyber attacks from occuring.

Link stability is a specific problem of link prediction that has been oftentimes overlooked as trivial. Although it shares the same set of domain challenges as link prediction, it does not predict future relations that may occur due to inferences from present observations. Instead, it ranks links shared between actors according to their structural importance to a community by their stability index scores.

There are several major limitations in the study of link stability in literature. First, many existing detection methods use the static node mechanism which fails to consider the intrinsic feature dynamics in the detection process. Additionally, most approaches are tailored to the use of a specific network in question and are not adaptable to more generalized social platforms. Furthermore, stable link identification is a largely unexplored area of research development without a structured framework of approach. This chapter will make scientific contributions to enhance the current detection capabilities of stable links to preserve structural integrity within a community and safeguard against detrimental effects of harmful, unstable external social influences.

In this chapter, we will present our MVVA (Multi-Variate Vector Autoregression) model for link stability detection, which is developed to encompass the multi-variate feature aspects of links in a single regression model. Its objective function bridges the gap between temporality and stability metrics. The scientific contribution of our work involves the following:

1. Our method bridges the gap between temporality and stability of links in online social networks. As an improvement to conventional static node and neighbor link occurrence methods, our approach is able to handle dynamic link features efficiently in the "prediction" process;

2. An innovative Hamiltonian Monte Carlo estimator is developed to help the MVVA model scale up to increasing dimensionality as the data volume grows arbitrarily large;

3. Experiment results show that the MVVA is able to offer a good modeling of the ground truth growth distribution of stable links within a Facebook clique with a good accuracy performance.

The rest of the chapter is organized as follows. Section 3.2 presents a brief outlook and overview of related work and literature reviews. Section 3.3 elaborates on the implemented methodologies and theoretical frameworks. Section 3.4 presents the results and discusses the analysis of the graphs and figures. Section 3.5 summarizes and concludes with a short indication of the future direction for the research work on link stability within the domain of structural integrity of OSNs and SISs.

3.2 Related Work

Social Network Analysis (SNA) has a long history based on key foundational principles of similarity. It has long been postulated that similar relationships between actors contain crucial information about social structure integrity [79]. The paradigm of link dynamics and their impact on structure is a question most social models struggle with solving. Furthermore, this has recently been made more complicated with the emergence of Heterogeneous Networks (HNs) and Social Internetworking Scenarios (SIS). In this section, we briefly review the state-of-the-art techniques and approaches of research done in two major areas of stable community and stable link detection.

3.2.1 Stable Community Detection

A community has intuitively been recognized by strong internal bonds and weak external connections. The measure of strength in connectivity has usually been represented by quantity over quality of connections within a group. These measures therefore, have always represented relational densities of varying scales. Thus, most clearly defined communities have always been often characterized by dense intra-community relationships and sparse intercommunity links at node edges [102], [105]. However, similar classical techniques have always suffered from several drawbacks because the detected community structure has not remained stable over time [93]. Detection of stable communities thus, requires the identification of stable links to serve as core structures of influence upon which a group of actors establishes online relations around [103].

In [209], a proposed framework to detect stable communities was developed. This was achieved by enriching the structure with mutual relationship estimations of observed links. In their study, link reciprocity estimation of backward edges and link stability scores were first established. The focus was given to detecting the presence of mutual links by preserving the original strength of backward edges, which scaled better with longer time observable windows. Stable communities were then discovered using the enriched graphical representation containing link stability information. This was done through a correlation of persistence probability (repeated time existence/occurrence) of each community and its local topology.

In [210], Charkraborty et al. studied how results from community detection algorithms change when vertex orderings stay invariant. By stabilizing the ranking of vertices, they have shown that the variation of community detection results can be significantly reduced. Using the node invariance technique, they defined constant communities as regions over which the structure remained constant over different perturbations and community detection algorithms over time.

3.2.2 Stable Link Detection

In [114], the authors suggested an activity-based approach to establish the strength (stability) of a social link. In contrast to friendship structures, their approach centered around a common disregarded aspect of activity networks. They argued that over time, social links can grow stronger(stable) or weaker(unstable) as a measure of social transaction activities. The study involved an observation of the evolutionary nature of link activities on Facebook. Their findings indicated that link prediction tasks relying on link occurrences as baseline metrics of measurements were inaccurate. As their results have shown, links in an activity network tend to fluctuate rapidly over time. Furthermore, the authors explained that decaying strength(stability) of ties correlate to decreasing social activity as the social network ages.

The study in [23] presented an overview of how links and their corresponding structures were being perceived from common link mining tasks. Such tasks

included object ranking, group detection, collective classification, link prediction and subgraph discovery. The authors argued that these techniques addressed the discovery of patterns and collections of Independent Identically Distributed (I.I.D.) instances. Their methods were focused around finding patterns in data by exploiting and explicitly modeling time-aware links among data instances. In addition, their paper contribution presented some of the more common research pathways into applications which were emerging from the fast-growing field of link mining like [211].

In summary, detecting stable links still remains an important aspect of many inference and prediction tasks which online applications use all the time [212], [213]. Community detection and link prediction are both concerned with identifying correlated distributions from a social scene [78]. These distributions can then be used as measures for decision support and recommendation systems [214].

3.3 Our Method

In this section, we detail our method for detecting stable links. The core of our model is developed from a regressional technique and was later refined to integrate with a stochastic approach for the cross-validation of accuracy and performance within a small Facebook clique.

3.3.1 Multi-Variate Vector Autoregression

The time series regression technique was chosen as the main approach to compute the stability index of links within a network. For small-scale datasets, vector regression methods(VAR) offer a very simple yet elegant means of analysis. Time series regressions are very simple and direct approaches. They are most often used in two forms to solve problems from a topological perspective. The first of these are the reduced (primary) form used in forecasting while the second is the structural (extended) form used in structural analysis.

In our work, we have adopted the structural framework as one of the core methods of approach towards identifying stability in links. Structural re-

gressions have the ability to benchmark relational behavior against known dynamic models in the social scene. It can also be used to investigate the response to disruptive surprises. Such social disruptions often occur as shocks from world events (e.g. the Brexit from the E.U., etc.).

A multiple linear regression model essentially extends the single regression model by considering multiple independent variable relationships to estimate the state of a dependent variable. MVVA extends this principle further by correlating the multi-linear regression relationships through time. Given a series of past dependent observables Y_τ, one can predict the unobserved dependent variable at the current time Y_t from the following mathematical formula:

$$Y_t = B_0 + \sum_{n=0,\tau=0}^{m,t-n} (G_n Y_\tau + \varepsilon_\tau) \tag{3.1}$$

Where B_0 is the array of residual constants and ε_τ is the error vector with zero variance co-variance.

Under the MVVA model which we have proposed, the six chosen variables of our study have been identified to be pivotal contributors of link stability. These identifications were studied from correlations, scatter plots and simple regressions between independent and dependent observables. It allows useful interpretation of observed relational behaviors which can be used for a variety of other tasks as well.

The stability matrix at time t is calculated from the predicted contributions of the six independent variables used in our study. We define the Stability index from Node Feature Similarity as $N(S)_t$, Cumulative Frequency as $F(Q)_t$, Sentiment as $I(S)_t$, Trust as $R(S)_t$, Betweenness as $B(S)_t$ and Transactions as $W(S)_t$. Thus, the stability contribution matrix S_t of all the six features is given as: $S_t = [N(S)_t, F(Q)_t, I(S)_t, R(S)_t, B(S)_t, W(S)_t]^T$.

From a structural perspective, the model we have developed follows the following mathematical formulation:

$$AS_t = \beta_0 + \sum_{\tau=1}^{p} (\beta_\tau S_{t-\tau}) + U_t \tag{3.2}$$

Where A is the restricted correlation matrix between the endogenous variables (dynamic feature stability contributions) identified through its past variations. β_0 and β_τ are structural parameters estimated through the method of Ordinary Least Squares (OLS). Hence, $\beta_\tau = A * G_\tau$. Finally, U_t are the time-independent disruptions caused by unsettling world events. This is derived from the (linear) system of equations as:

$$a_{11}N(S)_t + a_{12}F(Q)_t + a_{13}I(S)_t + a_{14}R(S)_t + a_{15}B(S)_t + a_{16}W(S)_t =$$
$$\beta_{10}+\beta_{11}N(S)_t+\beta_{12}F(Q)_t+\beta_{13}I(S)_t+\beta_{14}R(S)_t+\beta_{15}B(S)_t+\beta_{16}W(S)_t+U_{N(S)_t}$$

$$a_{21}N(S)_t + a_{22}F(Q)_t + a_{23}I(S)_t + a_{24}R(S)_t + a_{25}B(S)_t + a_{26}W(S)_t =$$
$$\beta_{20}+\beta_{21}N(S)_t+\beta_{22}F(Q)_t+\beta_{23}I(S)_t+\beta_{24}R(S)_t+\beta_{25}B(S)_t+\beta_{26}W(S)_t+U_{F(Q)_t}$$
.
.
.
$$a_{61}N(S)_t + a_{62}F(Q)_t + a_{63}I(S)_t + a_{64}R(S)_t + a_{65}B(S)_t + a_{66}W(S)_t =$$
$$\beta_{60}+\beta_{61}N(S)_t+\beta_{62}F(Q)_t+\beta_{63}I(S)_t+\beta_{64}R(S)_t+\beta_{65}B(S)_t+\beta_{66}W(S)_t+U_{W(S)_t}$$

In its primary form,

$$S_t = C_t + \sum_{\tau=1}^{m,t-n} G_\tau S_\tau + \varepsilon_t \tag{3.3}$$

Where, $C_t = A^{-1} * \beta_0$, $G_\tau = A^{-1} * \beta\tau$ and the residual errors $\varepsilon_t = A^{-1} * U_t$.

The number of independence restrictions imposed on the correlation matrix A is simply the difference between the unknown and known elements obtained from the variance co-variance matrix of the errors, $E(\varepsilon_t \varepsilon_t') = \Sigma_\varepsilon$. For the symmetric matrix of our model, $A = A^T$, which is $\frac{n^2-n}{2}$.

We define the feature rate coupling ratio w_t as the weighted impulse responses due to the structural disruptions on the endogenous feature observables. Each dynamic link feature response includes the effect of specific disruptions on one or more of the variables in the social system - at first occurrence t, and in subsequent time frames, $t + 1$, $t + 2$, etc.

The feature rate coupling ratio is thus given as:

$$\sum_{\tau=1}^{n} w_{U_\tau} = \sum_{\tau=1}^{n} (\dot{w}_{U_{\tau-1}} * [F_{U_\tau} - F_{U_{\tau-1}}]) \tag{3.4}$$

Where $\dot{w}_{U_{\tau-1}}$ is the first derivative response lag, which measures the momentum vector of social activity and F_{U_τ} and $F_{U_{\tau-1}}$ are endogeneous feature observable vectors at current and lag time frames respectively.

Then, we can express our structural autoregressive model in a vector sum of social disruptions as:

$$S_t^i = \mu + \sum_{i=0}^{k} w_{t,i} S_{t,i} \tag{3.5}$$

Where S_t^i is the stability matrix (with each feature element in i indicating how stable each link is). $w_{t,i}$ is the feature rate coupling ratio at time t and $S_{t,i}$ is the stability contribution; both across i endogeneous feature observables. Finally, μ is the impulse residual constant.

The MVVA model is not without its drawbacks. The complexity of the OLS problem involving a Cholesky decomposition of matrix M is at least $O(C^2N)$, where N is the sample data size and C is the total number of features. By direct inference, MVVA entropies to the squared growth in network complexity. Furthermore, two additional problems may arise as complexity of the social network grows; i.e. overfitting and multi-collinearity.

To overcome the above problems, we explore the Hamiltonian Monte Carlo (HMC) as an important extension to address the limitations of MVVA from a stochastic perspective for link stability detection. Since the social network we obtain from the repositories of common crawl contains missing links and partial information, stochastic estimations are used to measure the accuracy and reliability of our experimental MVVA results [215]. Additionally, HMC models are powerful samplers of potential energy distributions and its partial derivatives - which are representative of online social structures [31]. This means that overfitting and multi-collinearity will be tackled through high acceptance ratios [31]. Furthermore, the complexity per transition is O(GN). Where G is the gradient cost of the exact model which scales linearly with data and N is the number of steps [216].

3.3.2 Hamiltonian Monte Carlo

The condition that full form adaptive MCMC methods satisfy is:

$$\sum_{x} T(x' \leftarrow x)P(x) = P(x^i) \tag{3.6}$$

For a good sample x from the distribution P(x). x' is the next step-wise sample from x.
The Hamiltonian Monte Carlo extends the sampling efficiency of posteriors made by MCMC, through the use of Hamiltonian dynamics [217]. As an energy-based method, it is postulated that the sum total of all energies within a closed link-dynamics based system is conserved [71].

Hence, for every feature identified in the belief state graph G, its stability index score can be correlated to vector positional (static, potential) energy function $e^{H(G)}$ for any combinational variant of the graph $g \in G$ [72]. The Hamiltonian dynamics recognizes that a single form of energy cannot exist alone because it has to be conserved. Therefore, wherever potentials are the effects, the kinetics are the causals [217]. By introducing another variable which isn't our main information of interest, we are able to conserve this "relational energy" within the closed social belief system [178]. This can be identified as the transitional tensor (moving, kinetic) energy function $e^{-v^T v/2}$ between the different features and their states, such that this joint distribution is given as:

$$P(x, v) \propto e^{-E(x)} e^{-v^T v/2} = e^{-H(x,v)} \tag{3.7}$$

Where $P(x, v)$ is the conditional state transition probability between energy vectors x and v.

Firstly, the Leapfrog integration $L(\epsilon, M)$ is performed M times with an arbitrarily chosen step size ϵ. This means that $L(\zeta)$ is the final resulting state from M steps from the HMC dynamics with predefined step size ϵ. The next state transition step is given as:

$$\zeta^{(t,1)} = \sum_{n=1}^{k} L^n \zeta^{(t,0)} \text{ with probability } \pi_L^n(\zeta^{(t,0)}) \tag{3.8}$$

It is probabilistically defined as a Markov transition on its own [216]. The state transition momentum vector resulting from the secondary added accountable term for kinetic energy is then further corrupted by Gaussian noise

so that there are uncertainties during the transition of the states [218]. This is important because the non-deterministic nature of the momentum during transitions allow for proposals from current states onto new and further displaced states.

The randomization operator $R(\beta)$ mixes Gaussian noise determined by $\beta \in [0, 1]$ into the velocity vector given as:

$$R(\beta) = x, v' \tag{3.9}$$

$$v' = v\sqrt{1-\beta} + n\beta \tag{3.10}$$

Where n is drawn from a normal distribution:

$$n \sim N(0, I) \tag{3.11}$$

The transition probabilities are then chosen as:

$$\pi_{L^a}(\zeta) = min \begin{Bmatrix} \pi_{L^b}(\zeta), \\ \sum_{b \leq a} \frac{p(FL^a(\zeta))}{p(\zeta)}(1 - \sum_{b \leq a} \pi_{L^b}(FL^a(\zeta))) \end{Bmatrix} \tag{3.12}$$

Which satisfies the reversibility of the Markov Chain fixed positional transitional vector.

3.4 Experimental Results

In this section, we present the setup and results of our experimental evaluations on both MVVA and HMC algorithms.

3.4.1 Baseline Models

We consider several state-of-the-art methods for comparison with our proposed MVVA model. Since our model is developed from a stochastic approach, we use modified versions of similar methods along with the baselines we developed earlier for comparison. Another notable point is that although many prediction models exist, not all methods have the same goal or data features as ours. Therefore we consider only the models which use similar

data to ours for comparison. It should be mentioned that not all the methods can both predict links and profile link stability index together. Therefore we compare only the link stability prediction outputs between each other. The short description of the competing methods are given below:

Hub Promoted (HP) [219] is a topology based approach which compares the similarity of overlaps in the dense modular topological structure between node pairs which are sparsely linked together. In this approach, stability index is measured as a function of similarity between hierarchical modularity - which represents both clustering coefficient and the degree of modularity between node pair clusters. This baseline enables us to directly model structural perspectives while ignoring relational attributes that nodes commonly share. For the experiments, the scaling law [220] was used to quantify the node cluster hierarchy. The average-linkage hierarchical clustering algorithm [221] was also leveraged to determine the proximity of overlap between the clusters.

Resource Allocation (RA) [222] is a network topology metric which measures the common neighbours degree. This method is built around the estimation of contribution of common neighbours to the likelihood of the link prediction problem between node pairs in question. The higher the common neighbour node degrees, the more heavily penalized will their contributions be towards this expectation. In this experimental baseline, common neighbours and their degrees are used to calculate the RA metric. The RA score is then used as a function of prediction - to rank stability scores of links between node pairs of the dataset.

Cosine Similarity Time (CST) [223] is a path based method used to measure both path and temporal similarity in the number of hops required to transition between node pairs in a given OSN. This baseline method leverages on the cosine similarity measure of the temporal vectors in a pseudo inverse adjacency matrix space. For this baseline model, key vector paths between selected node pairs are extracted. Using metrics from both Hitting Time (HT) and Commute Time (CT), the Stability Index (SI) is then characterized, through the use of normalized CST scores. The higher the CST score of a symmetric pathway measured between node pairs is, the more stable the

links forming the pathway in question will be.

Restricted Boltzmann Machine (RBM) [114] is a shallow Artificial Neural Network (ANN) built from a single confabulation layer. This baseline method directly accepts inputs from dataset features and adjusts weights in between the confabulation for proper visible layer representation of link stability at the output. As a fully supervised process, this method first starts with a random initialization of values for the training feature dataset. Then secondly, as the RBM is successively trained for each input extracted from the dataset, outputs are constantly adjusted and fed back into the system as expected SI values based on a function of the Node Feature Similarity Index.

Deep Belief Network (DBN) [114] is a Deep Neural Network (DNN) composed of a multi-layer stack of RBMs. This baseline approach is semi-supervised and leverages on multiple representation layers to derive an output for link Stability Index. In the experiments that follow, important expectations for link Stability Index at the output is adjusted at intervals during the training process. These temporal feedback interventions are based on the Truth Values selected for this study. The semi-supervised approach allows the DBN model, the autonomy to derive outputs of SI based on key correlations at the input with temperance from human supervision at intermissions in the process.

Node Feature Similarity Index (True Value) [23] This node based metric is used as ground truth values against the models prediction. The relational features between node pairs are compared. Similarity scores are calculated based on cumulative frequency, content sentiment, node betweenness, trust reciprocity and the number of posts. These are features which are obtained from the dataset used in this study.

Multi-Variate Vector Auto-regression(MVVA) is the model which we have developed in this paper that analyses relational features in a single regressional model. Using the stochastic model, key relational features of interest between nodes are applied as inputs to uncover auto-correlations. The strength of correlations between the independent input features are then regressed to predict the stability (dependent variable) of the link between nodes

at the output.

3.4.2 Experimental Setup

The dataset chosen for this study was crawled from Facebook and obtained from the repository of the Common Crawl (August 2016) [1]. It includes the following relational features between any two arbitrary nodes: The Cumulative Frequency of the type of wall posts, the sentiment of the content in context of the post (Neutral, Positive, Somewhat Positive, Mildly Positive, Negative, Somewhat Negative, Mildly Negative), the Node-betweenness Feature Similarity (Roles and Proximity metrics), the Trust Reciprocity Index (Similar in quantization to Sentiment Index) and the number of posts at defined quantized Unix time sample space as a measure of link virility. In this study, the Node Feature Similarity Index is used as a performance benchmark against multivariate analysis.

The experiments were conducted on our Multi-Variate Vector Auto-Regression Model on undirected small world topologies with a clique size of 20-100 nodes. A subset of nodes (< 10) was first chosen for this study as the defining seed community. Then, this chosen community was allowed to grow to a maximum size of 1019 nodes by adjoining nodes to establish new relationships.

The links in the network are tagged based on their Stability Index (SI) scores. Stable links are labeled with SI scores higher than or equal to 80, while Neutral links are labeled with SI scores in the range of [50,80), the slightly unstable links are labeled with SI scores in the range of [30,50) and the unstable links are labeled with SI scores in the range of [0,30) respectively. The new and existing links which are SI score labeled (satisfying their respective threshold conditions) were then subsequently evaluated for their Aggregated Link Stability Index over time at every sample (whenever social transactions were captured by the crawler across posts) based on the variate features discussed above. The aggregated link stability index is calculated as:

$$AG_t = \sum_{i,E=0}^{k,m} S_t^{i,E} \tag{3.13}$$

[1] https://commoncrawl.org/2016/09/august-2016-crawl-archive-now-available/

Where AG_t is the aggregated link stability index of the topograph at time instant t and $S_t^{i,E}$ is the feature i stability index of edge E in the network.

The prediction error is given simply as:

$$e_t = |Y_t - F_t| \tag{3.14}$$

Where e_t is the Aggregated Stability Index Prediction error, Y_t is the observed Aggregated Stability Index at time t - this is given by the HMC Stability State Index values after a 100 times iteration over the samples of the 5 multivariates. F_t is the predicted Stability Index based on both the MVVA model and the univariate (Similarity Index) regression model.

The scaled error across the two different datasets is given by the equation as:

$$\varepsilon = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|} \tag{3.15}$$

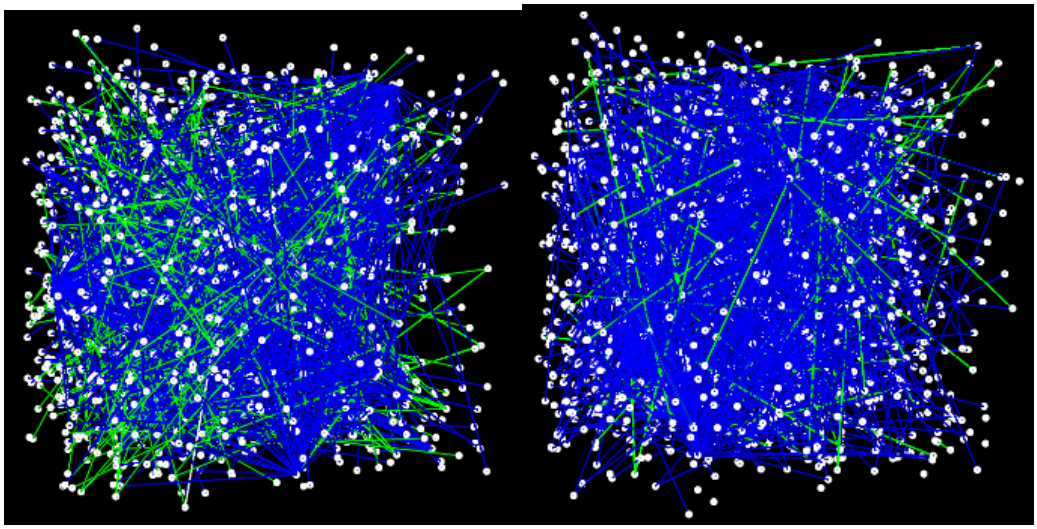
Where ε_t is the absolute scale free error of the predicted data set F_t against the observed dataset Y_t. n is the number of sampled forecasts. The Mean Absolute Scaled Error (MASE) of a distribution plot Q is given as:

$$MASE(Q) = \sum_{t=0}^{k}\frac{\varepsilon_t}{k} \tag{3.16}$$

3.4.3 Link Stability Evaluation for MVVA

In our experiment, the link growth comparison is conducted between the univariate node based Common Neighbor (CN) feature, the univariate features (which includes Similarity Index as well) and the model baselines. As seen in Figure 3.2, by considering the dynamics of the relational features within those established links in the multi-variate time regression process we have proposed, our Multivariate Link Stability Index outperformed the CN-Node Similarity based Stability Index by over twice the score of the traditional metric used in the link prediction process with an AUC of 0.87 by comparison to the latter's AUC of 0.46; which is a tremendous improvement in terms of efficacy. Furthermore, models from our selected deep network approaches [114] also agree quite well with the growth profile of our proposed MVVA approach. The number of labeled links of the fully evolved topograph at

Figure 3.1: Topograph of Univariate Similarity based Stability Index (left) and Topograph of the Multivariate Stability Index (right)



(a) Univariate Similarity Model (b) Multivariate Similarity Model

This diagram shows that measuring stability index of links in an OSN, lesser but more reliable links are detected as stable using the multivariate (MVVA) approach while more non-reliable links are detected as less stable using univariate approaches.

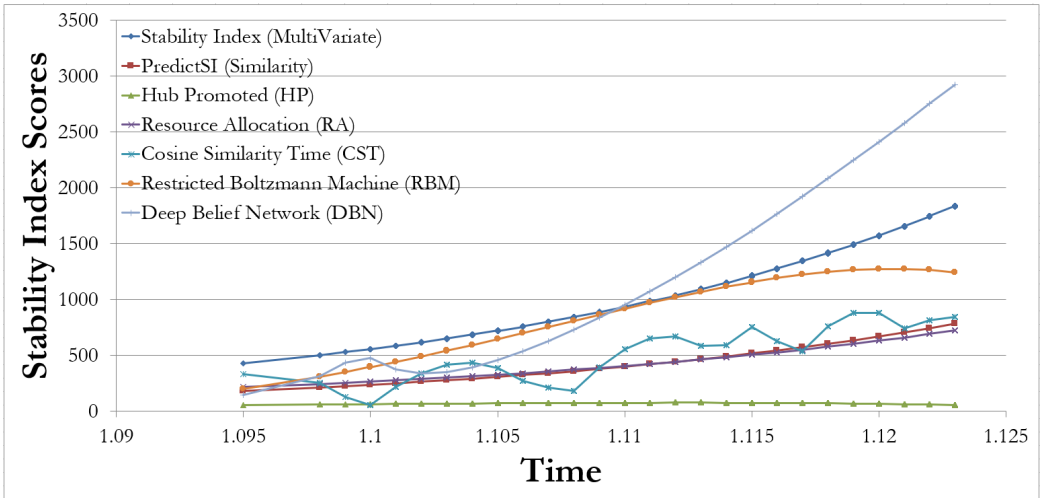


Figure 3.2. Link Stability Index comparison over time: both scales (Stability Index Scores and Unix Time) have been normalized to fit into the plot frame window.

the end of 30 days and the calculated aggregated stability index are given in Table 3.1 and 3.2.

Table 3.1: Tabulation of the number of identified labeled links for both Univariate and Multivariate Regression Analysis

Type of Link	Univariate	Multivariate	HP	RA	CST	RBM	DBM	Score Range
Stable	3184	694	2899	2749	2294	871	417	> 80
Neutral	5257	7782	5012	4913	4792	6924	7133	50-79
Somewhat Stable	35	0	273	388	428	632	693	30-49
Unstable	0	0	292	426	962	49	233	0-29

Table 3.2: 30-day Normalized Aggregated Stability Index

Multivariate	1835
Univariate	783

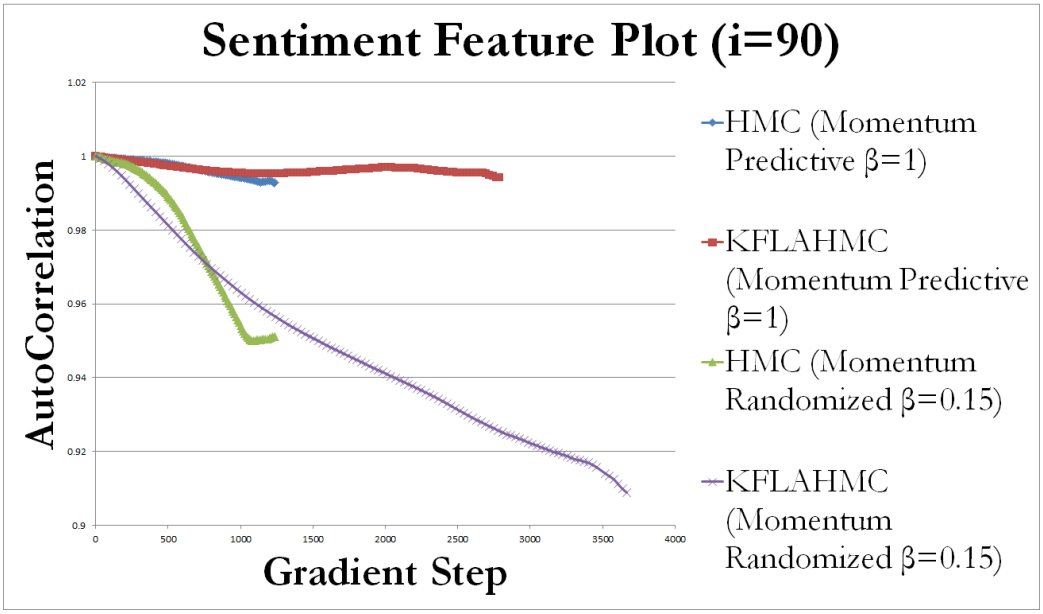


Figure 3.3. Graph of Sentiment Autocorrelation against the number of gradient iterations for predictive ($\beta = 1$) and randomized ($\beta = 0.15$) momentum vectors of HMC for 10 burn in data sets of the similarity feature from the Facebook wall posts.

Figure 3.1 shows the growth distribution of the stability index scores of links within a Facebook clique for a period of 30 days. The experiment was done using the MVVA autoregressive algorithm for both univariate and multivariate modes of calculation, of the dataset acquired from Facebook. It measures the aggregate stability scores accumulated within the clique against the time - which has been normalized to fit into the scale window of the plot.

3.4.4 MVVA Accuracy Evaluation

Based on Figure 3.2, Table 3.1 and 3.2, it can be seen that stable link detection accuracy using our model has been vastly improved by 78.29% with 3184 links being detected as stable in the univariate analysis; and only a similar 694 links detected in the multivariate analysis. Furthermore, with only 694 links identified as stable in the Multivariate Analysis, the aggregated scores of the topology are 2.34 times higher than the Univariate Analysis; suggesting a noticable improvement in terms of efficacy - making our model far more reliable than traditional univariate methods throughout the prediction

	MVVA		Univariate	
	In	Out	In	Out
MASE	0.074268	0.0944732	0.616677	0.572323

Table 3.3: Tabulation of Mean Squared Errors (MASE) of both Multivariate and Univariate Analysis at the end of the 30-day clique evolution period.

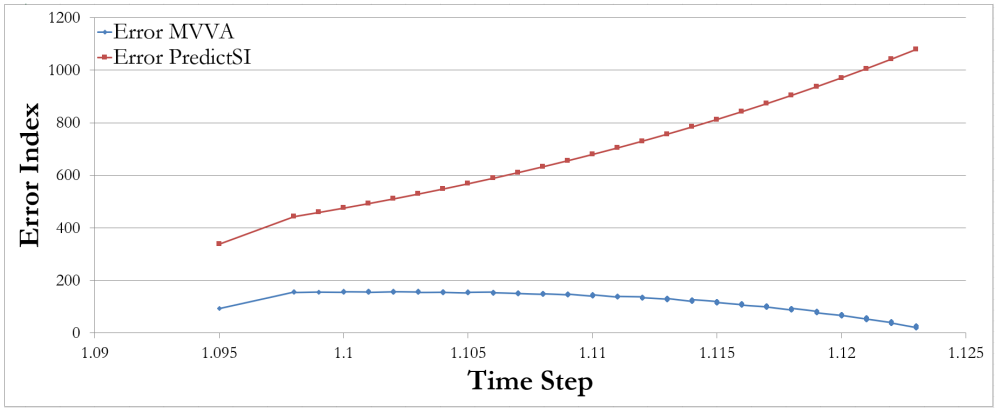


Figure 3.4. Error score ϵ_t comparison over time between MVVA and the univariate regression models.

process.

3.4.5 Prediction Error Evaluation

The prediction error results can be summarized in Table 3.3. As can be seen from Figure 3.4, the error score index ε_t grows over time for the univariate regression analysis, whereas the error score index ε_t of the MVVA model which we proposed decreases over time. Additionally, as can be seen from Table 3.3, the MASE score for the MVVA model improves both the In-Sample and Out-Sample prediction accuracy of the underlying stability index distribution for the Facebook clique over the 30-day time frame by 8.3 times more than the MASE score for the conventional univariate regression model.

3.4.6 HMC Results and Evaluation

Figure 3.3 shows good (small) autocorrelations between the training data of features in most sets, although there are some sets which present spurious/biased information where a Gaussian distributed and noise-corrupted momentum sampled model could not correlate well to with respect to log distributions of its momenta and positional gradients. However, it can be seen that from more burn in data samples and more randomized (corrupted by noise - $\beta = 0.1$) momenta sampling behavior, the performance of the gradient autocorrelation improves during the learning phase of our HMC implementation.

Figure 3.5 is a posterior sample of Sentiment index scores. The horizontal axis reflects the normalized time which has elapsed during the process and is also directly proportional to the number of iterations progressed through this window (as displayed on the graphs).

Figures 3.6 to 3.9 show progressively how the random walk proposed distribution converges towards the actual distribution of the Stability Index data set from a fixed point condition (the very first initial feature belief state at t=0) being held constant. Figure 3.9 is the Monte Carlo approximation for the actual 30-day aggregated stability index distribution repeated over Hamiltonian dynamics for 100 cycles. It shows a good convergence towards our MVVA model; which reflects very closely to the actual growth of aggregated stability index over time - as opposed to univariate (similarity feature) based link stability prediction.

3.5 Conclusion

In conclusion, the Multivariate model (MVVA) which we have proposed for the detection and identification of stable links works well and is far more superior to univariate models or models which consider only static node based features and link temporality. Our system has been tested on a small Facebook clique which was evolving. This dynamic growth can now be better understood and comprehended through the existence of stable links as other seed clusters form around it. However, the tighter, more stringent constraints

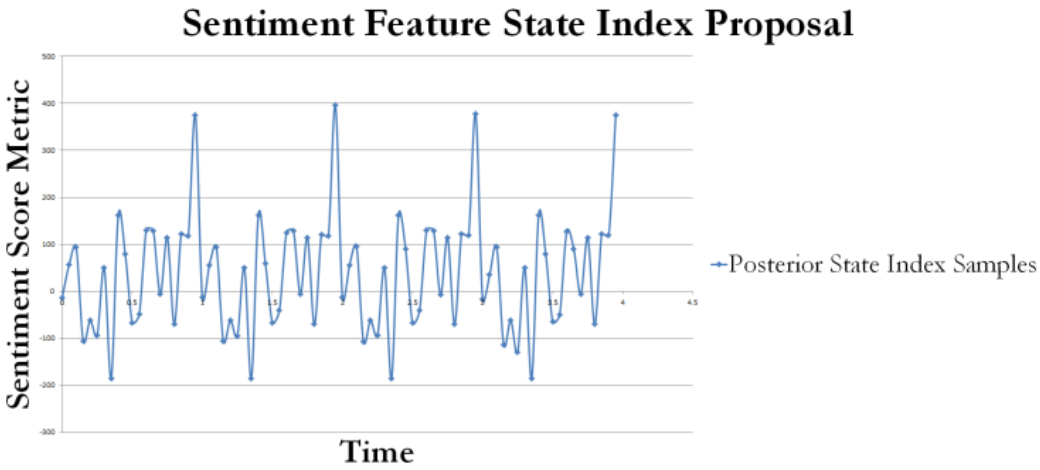


Figure 3.5. Plot of Posterior Sentiment Feature state samples. This plot shows the profile of sentiment feature sampling states from posterior distributions.

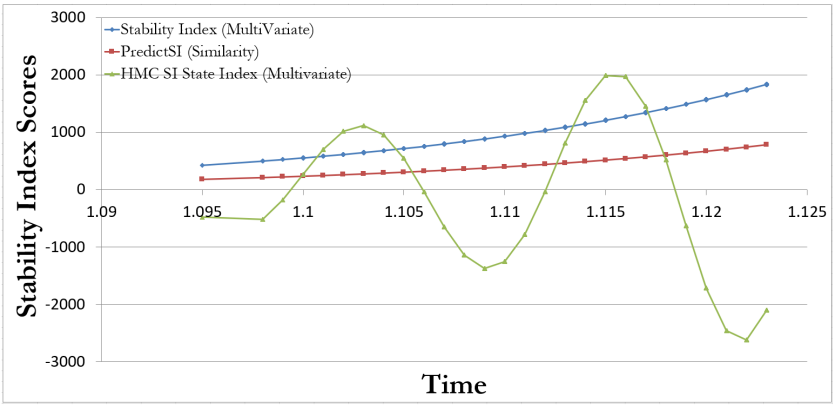


Figure 3.6. Link Stability Index comparison over time with HMC iterated over 10 times for posterior states of the 5 multivariates (Time Delta, Frequency, Similarity, Sentiment and Trust).

of a small world model used in this study should not be overlooked. In larger hyper-graphical models, where boundaries fall apart due to sheer volume distributions of scattered data, a larger scope of stochastic lemmas surrounding both high complexities and large volumes of social features have to be rediscovered [224].

Some advantages of our methods and experimentation include a strongly

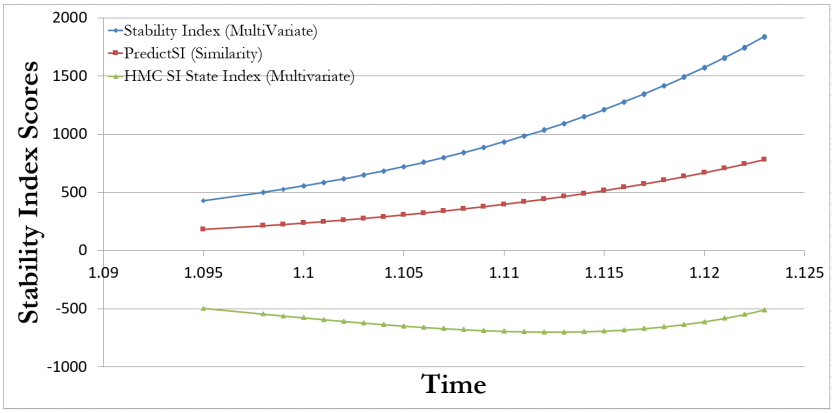


Figure 3.7. Link Stability Index comparison over time with HMC iterated over 50 times for posterior states of the 5 multivariates (Time Delta, Frequency, Similarity, Sentiment and Trust).

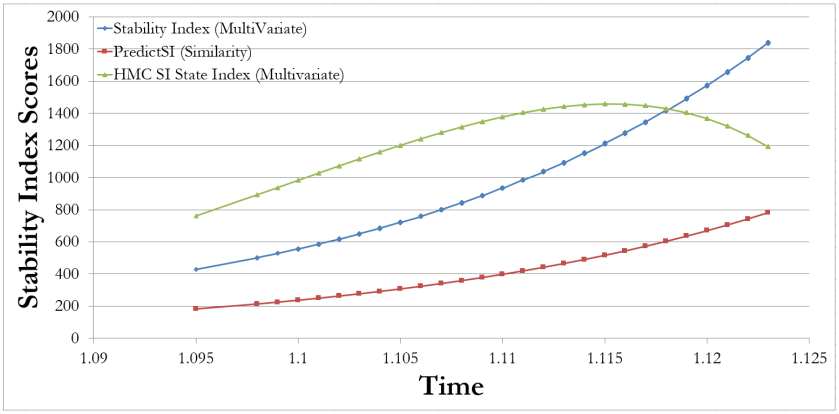


Figure 3.8. Link Stability Index comparison over time with HMC iterated over 80 times for posterior states of the 5 multivariates (Time Delta, Frequency, Similarity, Sentiment and Trust).

connected network with a firm belief structure and sufficient access to new information being made readily available during the data mining process. However, in larger dimensional frameworks where the constraints of such structure break down and data is made even wider and more sparse, deep

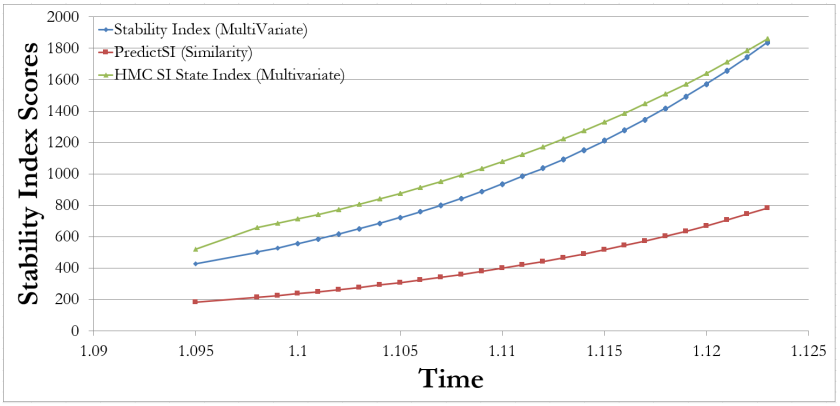


Figure 3.9. Link Stability Index comparison over time with HMC iterated over 100 times for posterior states of the 5 multivariates (Time Delta, Frequency, Similarity, Sentiment and Trust).

learning knowledge discovery methods like Monte Carlo estimates and the DNNs are powerful variations which can be used for online social prediction and inference tasks [77].

CHAPTER 4

IDENTIFYING RELATIONAL FLUX AND TURBULENCE

4.1 Introduction

Structural stability in social networks has always been a topic of contention in various applications of interest. These include but are not limited to link predictive approaches, community detection methods and logical random graph models [6]. The key elements of relational stability have always been referenced to attributes perceived to be contained within links established between key actors / nodes within a social community structure [7]. Recent studies performed in this area of interest also include the use of directional dyads and signed reciprocity as a special measure of link "strength" representation [225]. Amongst the multitude of relational approaches administered - to solve for a social objective function; many techinques however, still rely on "flat" uni-directional linked structures. The obvious drawback of using such a method is that key relational attributes shared between actors have been left out of the solution - thereby leading to inaccuracies and inconsistencies (instability) in the detected / predicted social structures and / or sub-structures [10]. A key observation that can be made through literature revolving around relational intelligence of a social network is an over-reliance on similarity measures between node to node or link degree features [11]. Although studies based on feature similarities have shown efficiency (quality) improvements in detection thresholds, and prediction patterns in comparison with other methods of interest; much of these approaches however, lack the representational efficacy to model real life social structures [16].

Prime examples of such studies done in the recent past are node-based measures like node feature similarity and text feature similarity, neighbor-based measures like Common Neighbors (CN), Jaccard's Coefficient (JC), Adamic

Adar (AA), Leicht-Holme-Newman Index (LHN), Preferential Attachment (PA) and Resource Allocation (RA), path-based measures like the Katz constant, LP, RSS, Friend link, Blondel Index (BI) and VCP, random walk-based measures like SimRank, PropFlow, Rooted Page Rank (RPR), Hitting Time (HT), Commute Time (CT), etc. [22]. These methods all rely on correlating observations between one feature to another - between two or more relational entities in question. Furthermore, the use of social theories has also been rather limited and crude within this field of study as well - with the emergence of dyadic and signed relational links in an attempt to represent complex dimensionalities like status, trust, influence, etc. Such representations capture relational structures from a time static perspective and are not as adapt to change as time-aware approaches that most recent studies into OSNs model after [23].

In this study, we introduce a new model - the Relational-Flux-Turbulence (RFT) that effectively represents the dynamism of popular key relational dimensions uncovered from previous approaches and techniques conducted on online social structures as a time evolving flow of relational attributes (time-realistic relationships) between node entities of a network in question with the constant inception of social shocks. The model builds a multi-stage deep neural network from a stack of fractals with hybrid architectures of Restricted Boltzmann Machines (RBMs) and Recursive Neural Nets (RNNs). These structures are self-evolving from a meta-learning perspective. The neural network accepts as inputs, key relational feature states f_i between actors a_j and global events E_ϵ from past and present social transactions to determine the likelihood of relational turbulence τ_{ij} within an identified social flux F_ϵ. Turbulence may correspond to various disruptions in social communication of different environments and contexts. For example, in the discussion of world events like trade wars, passive sentiments passed through public posts and comments are indicative of hostility and potential conflict which may lead to a breakdown of linked integrity between actors in many aspects like trust, influence, status, etc. In addition, as a major contribution of this chapter, we also show that time evolution relational flows improves efficacy of existing approaches through studies and comparisons of experimental results conducted on real life social networks. We develop a novel architecture from Relational Turbulence Theory and Models (RTT and RTM) to iden-

tify social disruptions by predicting the occurance of relational turbulence, within a given social context describing the state of flux. Then, we evaluate our methods on Twitter, Google and Enron email datasets and demonstrate that they outperform similarity based feature and shallow uni-directional flat structural approaches in detecting social flux and turbulence.

Data dimensionality reduction comes at a cost [74]. Such costs have often been quantified in terms of loss, errors, performance, etc. in common literature [226]. Neural Networks offer an alternative to learning in which such costs can be specified to a given margin - depending on the expected performance [162]. However, these costs unfortunately cannot be eliminated within a given model of choice. They can only be passed on from one index of measure to another. For example, in Deep Neural Network (DNN) architectures, if we wish to optimize prediction accuracy from a given bag of training data batches, we decrease the error tolerance at each learning epoch. What this effectively does is to pass on the gains in decreased measures of training errors over to costs of decreased performance. This means that if we wish for the model to predict more accurately at each epoch, we have to compromise that improvement in expectations with a decrease in convergence rates. A highly accurate DNN model therefore, requires long lengths of training time regardless of training bag sizes [163].

Regardless, many have justified these increases in costs to be reasonable measures of gains with the surge in advancements of high performing CPUs and GPGPUs [227]. The exponential increase in traditional Floating Point Operations per second (FLOPs) that current processing units are able to compute have helped absolved a tremendous portion of such training losses. However, speed and accuracy are not the only two mediums through which costs are being passed over. To date, many highly complicated and ultra-deep learning architectures like Deep Convolutional Networks (DCNs) [228], Long Short Term Memory modules (LSTMs) [192], Residual Networks (ResNets) [199], etc. have evolved from the need to represent increasingly highly sophisticated convolutions of real-world information like human speech, language, vision, relationships, etc. The improvement in computational power is constantly squared off by the increasing use of more complicated and complex learning architectures.

For a given set of hard constraints where both speed and accuracy are important factors to achieve an expectation threshold, information representation suffers [229]. This means that in order to reduce dimensionality of data without compromises on learning speed or prediction accuracy, information must necessarily be lost in the process. What this translates to in the modeling of progressive learning architectures (e.g. Progressive Neural Networks (PNNs) [230]) are that DNNs cannot be relied on to reproduce the results as and when required. Instead, shallow architectures have to be utilized in order to meet such demands. Hence, a critical unresolved question remains. How can be preserve the same levels of information sophistication learned from highly complex models in shallow Artificial Neural Networks (ANNs)? The answer lies in encoding knowledge dimensionality into a fully self-functional and highly volatile shallow ANN architecture which can be used to either generate or collapse depth complexity during learning - in response to random "anytime-sequenced" data batches of fluctuating information sophistication.

We study the dynamic structure of such an shallow ANN known as fractals. Fractals are the lowest principle decompositions of never ending patterns. They maintain a key property of self-similarity across different varying scales [26]. A careful distinction here requires an aggressive discrimination between similar and identical schemes. While fractals may maintain similar structural properties under any representational construct, rarely - if ever at all are they exactly identical in any one of them. They can grow to become complex enough to represent high levels of sophistication that are yet trivially efficient enough to re-create by repeating similar simple shallow architectures in a loop - ad infinitum. Driven by a recursive process, fractals are adaptable enough to describe highly dynamic system representations[18]. In the sections that follow, we describe the methods and experiments performed on Twitter, Google and Enron email datasets at different instances and show that structural fractals behave like cognitive super primers that can be used to decode representational information sophistication through a generative feedback loop. The main contributions of our study are:

1. Our method adaptively learns from real-time online streaming data to identify key turbulent relationships within a given OSN;

2. An innovative RFT model was developed to capture key relational features which were used to detect and profile social communication patterns of eventful states within a given OSN;

3. Experiment results show that RFT is able to offer a good modeling of relational ground truths, while FNN is able to efficiently and accurately represent evolving relational turbulence and flux profiles within a given OSN.

The remaining part of the chapter is organized as follows: Section 4.2 presents a brief overview of related works and introduces key concepts drawn from social theories and relational structures. Section 4.3 introduces the theories and methods of our proposed model. Section 4.4 provides a thorough analysis of experimental design and implementation. Section 4.5 presents the results and discussion of this chapter that leads to a conclusion and potential future directions.

4.2 Related Literature

Relational Turbulence was first studied in [231]. It was typically characterized as a resultant state in conflict of interests from competing goals between two or more actors in question [232]. Although conflict does provide the basis of stimulation for communication within a relationship that is centered in a flux, it also correlates to negative consequences in the form of detrimental event occurrences if left undetected and unchecked [233]. An important discriminator of detecting conflict and hence the resulting turbulence in any relationship model between networks of actors is the observation and management of relational altering events. These events, if found to be in huge negative violations of expectancies between relational reciprocates of actors, can lead to instability in a relational flux [234], [235].

Excluding relational expectation management, some detrimental relational altering events include: geographic displacements (or low proximity measures), conflict escalation (high frequencies of friction), environmental changes (expectation disparities), etc. [232], [236], [237]. Relational Turbulence is

briefly defined as changes which occur within a relationship that may cause friction [15] between actors and their local online community. These changes are often studied as a series of transitions (often abrupt) between actor-environmental states that inadvertently influences relational characteristics by changing communication flux patterns [238] of a given relationship in an Online Social Network (OSN). These shifts in relational characteristics during difficult state transitions (altering events) may lead to volatile consequences.

The Relational Turbulence Model (RTM) [20] builds upon the core principles of relational state shifts and conflict management to define an artificial construct. This construct enables intelligent predictions of communication behaviors during relationship transitions in an environment of continuous online social disruptions. The process of turbulent relationship development can be described as a continuous and communicative state of flux [20]. This state defines a consistent exchange of sentimental and affective information between the actor/s involved. Each transition to another state (e.g. professional colleagues to friendship) has the probability to cause friction (conflict), which may lead to a polarization of sentiments and affective communication flux in OSNs [9]. Two key tenets of the RTM are actor interferences and relational uncertainty. These two prime relational features in OSNs enable the effective detection and prediction of conflict and event occurrences in sentimental and affective computing.

While RTM explains and predicts relational conflict through communicative behaviors between actors, Relational Turbulence Theory (RTT) [9] correlates uncertainty and interference to specific behaviors, actions and sentiments (either hidden or expressed). A study covered in a later chapter looks into the theory of conflict mechanisms to build a Fractal Neural Network (FNN) from relational fractals - that can be used to predict events based on the principles of Relational Turbulence Theory.

In [239], the authors present a minimalist neural network architecture for reliably and accurately estimating emotional states based on EEG captured data. Their model uses an innovative parameter known as the reinforced gradient coefficient to tackle the vanishing gradient problem faced by deep learn-

ing architectures. Additionally, their model adopts a weighing step to extract outliers from the discrepancies between successive predictions. As a shallow Artificial Neural Network (ANN) architecture, their model aggregates two-hidden-layer independent confabulations which classify proportionately according to the cardinality of emotional states in question. Their weighted mean minimization function regularizes the Euclidean disparities between weight matrices of the hidden neural network layers in the model to circumvent the common problem of overfitting the output at the inputs. Although adapt at finding an optimum markovian leap step $l^{k,t}$ of the k^{th} classifier block at the t^{th} training iteration, their model suffers from a lack of representation for more deeply complex emotional states (e.g. an in-betweeness in quantization across valance and arousal). Additionally, their reinforced gradient coefficient merely augments the errors calculated between expected-weighted and actual outputs which are then used to update the layered weights of their shallow ANN model. Although this approach may help alleviate diminishing gradients by increasing error gradients during the back propagation process, it does so at the expense of performance. Tackling larger error gradients during the forward and back propagation burn-in phases of training especially on a shallow ANN architecture means that convergence to an accurate estimation is slower with more lengthy iterations. Furthermore, the trade off in accuracy gains between the MNN and other state of the art methods (e.g. ADA, RMS, NM, etc.) included in their work does not justify the computational resource costs involved.

In [70], the authors deal with the problem of social role recognition through the use of a Conditional Random Field (CRF) layered model architecture. Their architecture is used to learn actor-environment and actor-actor behaviors from different unlabeled video streams of a given event classification. Their work derives from the motivation in the field of Role Theory in sociology. This theory correlates identification and explanations to observations of behavior patterns classed under their respective social roles in question. Knowing social roles may help predict social interactions between actors within a certain eventful setting. Their social role discovery approach is defined as a weakly supervised problem. As a shallow layered learning architecture, their model is suitable for capturing unary features such as object interaction features and social features. Their algorithm inference

mechanisms rely on pairwise interaction features like Proxemic Interaction Features (PIF) and Spatio-Temporal Interaction Features (STIF). Their full model results on You-Tube social videos show a higher event-based social role classification hit-rate as compared to traditional K-means and CRF cluster algorithms. However, for video image frames in which latent social role-based semantics exists, CRF architectures are ill-adapted to handle the complex representations of the depth of these roles in the identification process. This invariably leads to poor performance output measures of their full model method.

Building on the principles of Role Theory, [69] proposes a deeper hierarchical model for human activity recognition based on identified actor roles within an eventful context. Their model captures the actor-environment-actor relationships within a given classified contextual reference of occurrence. These relationships capture and describe the dependencies and interactions between low-level actions, mid-level social roles and high-level events in any given scene. Their model builds on a Multi-Layer Perceptron (MLP) architecture in which social features and actor-environment actions act as lower layer inputs into the architecture that forward propagate into social interactions and scene-level event at the outputs. Their model formulation uses two labels as outputs which are associated to their respective hierarchical model representations: Action Models and Unary Role Models. They train their model parameter to produce a correct hierarchical structural event using a structured Support Vector Machine (SVM) approach which includes social roles, events and actions. This is done using a discriminative max-margin framework. Although their model is able to produce better precision-recall rations in a Pearson correlation test; for larger event frameworks however, this model may easily degrade in terms of performance due to problems of overfitting and error gradient saddle points.

In [240], the authors present experiments on the automatic recognition of roles of participants in meetings. Their approach aggregates lexical choices made by participants of the meeting and social interactions between actor-environment-actor scenarios. Their approach proposes learning of behavioral cues as a set of parallel paths which model lexical choices and interaction patterns as participant role features respectively. Their experiments were

performed over the AMI corpus and have shown a 70% hit rate in accurately predicting and identifying participant roles in a meeting. However, their method does not scale well to unpredictable and complex role behavior of participants at a meeting due to its flat learning architecture design.

4.3 Theories and Methods

From the RTM approach, we define Relational Intensity $P(\gamma_{rl})$, Relational Interference $P(\vartheta_{rl})$ and Relational Uncertainty $P(\varphi_{rl})$ to be three key probabilistic outputs of the RFT model which represents the relational turbulence $P_{\tau_{rl}}$ of a given link in an OSN. 1. The key element types we have identified to be contributing features between the duration of the turning point and relationship development (as an unstable / turbulent process) are the Confidence ρ_{ij}, Salience ξ_{ij} and Sentiment λ_{ij} scores in an actor-actor relationship of a social transaction in question. It is noteworthy of mention that the ground truth reciprocities of these element types shared within a relational flux, violates expectancies - $E(\rho_{ij})$, $E(\xi_{ij})$ and $E(\lambda_{ij})$ respectively [9]. These violations (however small), are a contributing factor to temporal representations of relational turbulence - γ_{rl}, ϑ_{rl} and φ_{rl}. 1. Negative expectancy is defined as a polar mismatch between expected reciprocates against actual reciprocates (e.g. Actor i expecting a somewhat positive reciprocation of an egress sentiment stream, but instead, received a negative ingress sentiment stream from actor j). Positive expectancy is defined as the strong cosine similar vector alignment between these reciprocates. Both expectancy violation (EV) extremes however, are characterized by sharp gradient changes of their weighted feature scores. This is given mathematically as:

$$\frac{\partial E_{rl}}{\partial \tau_{rl}} = \sum_{i,j=1}^{n} \prod_{\eta=\rho,\xi,\lambda} \frac{E(\eta_{ji})}{\partial \eta_{ji}} \times \frac{\partial \eta_{ij}}{\partial \tau_{ij}} \tag{4.1}$$

Where τ_{ij} is also known as the relational turbulence between node i and its surrounding neighbors j.

Relational change or transition also known as a turning point, defines some state-based critical threshold, beyond which relational turbulence and negative communication is irrevocable [21]. This critical threshold is specific

to actor-actor relationships and learned through our model as a conflict escalation minimization function [241]. Conflict escalation is defined as the gradual increase in negative flux $\frac{-\nabla F_\epsilon}{\nabla t}$ over time within a classified context area L_{F_ϵ} of interest [241]. The critical threshold parameter is then driven mathematically as:

$$T_\epsilon = inf_{t\to\infty} \begin{cases} \frac{1}{2m}(-\frac{\nabla F_\epsilon}{\nabla t} \times \log_2(\frac{\nabla F_\epsilon}{\nabla L_{F_\epsilon}})) \\ \log_2(|1 - \frac{\nabla F_\epsilon}{\nabla L_{F_\epsilon}}|) \end{cases}$$

Where T_ϵ is the threshold of interest and m is the total number of training data over the time window t.

Firstly, we define relational intensity during state altering events conditionally, as the integration of sentimental transactions (flux) per unit (context) area. Mathematically, this is given as:

$$\gamma_{rl} = \sum_{i,j=1}^{n} \frac{\beta_{ij}|-\frac{\nabla F_{\epsilon j}}{\nabla t}|}{L_{F_\epsilon}} \tag{4.2}$$

Where β_{ij} is defined as the temporal derivative of the latent topic (context) signal phase ϵ. Secondly, we define relational uncertainty as the likelihood measure of opposing sentiment mentions. Mathematically, this is given by:

$$\varphi_{rl} = \frac{\sum_{i,j=1}^{n} S_i S_j}{\sqrt{\sum_{i=1}^{n} S_i}\sqrt{\sum_{j=1}^{n} S_j}} \tag{4.3}$$

Where S_i and S_j are sentiments transacted from nodes i to j and from nodes j to i respectively. Finally, we define relational interference as the probability that deviations from predicted or expected outcomes of relational flux intensity and uncertainty fall outside a confidence interval centered about the mean. Mathematically, this is given as:

$$\begin{aligned} \vartheta_{rl} &= E(F(\gamma_{rl}, \vartheta_{rl} : \mu_{\gamma\varphi}, \omega^2_{\gamma\varphi})) \\ &= \tfrac{1}{2} + \tfrac{1}{\sqrt{2\pi}\omega} \sum_{\gamma_{rl},\varphi_{rl}=0}^{n} \tfrac{1}{2} erf(\tfrac{\gamma_{rl},\varphi_{rl}-\mu}{\sqrt{2}\pi}) \exp^{\frac{-(\gamma_{rl},\varphi_{rl}-\mu)^2}{2\omega^2}} \end{aligned} \tag{4.4}$$

Where,

$$F(\gamma_{rl}, \varphi_{rl} : \mu_{\gamma\varphi}, \omega^2_{\gamma\varphi}) = \frac{1}{\sqrt{2\pi}\omega} \sum_{t=-\infty}^{\gamma_{rl},\varphi_{rl}} \exp^{\frac{-(t-\mu)^2}{2\omega^2}} dt \tag{4.5}$$

Here, $F(\gamma_{rl}, \varphi_{rl} : \mu_{\gamma\varphi}, \omega^2_{\gamma\varphi})$ is the cumulative distribution function, and $erf(x)$ is the error function of the predicted outcomes γ_{rl} and φ_{rl}.

The model we have chosen, to address the prediction problem of relational turbulence is the Fractal Neural Network (FNN) that adopts a hybrid architecture which incorporates the use of both generative and discriminative deep networked architectures. At the core of the RFT architecture is a stack of Restricted Boltzmann Machines (RBM) which constitutes the essence of a Deep Belief Network (DBN) that pretrains our Deep Neural Network (DNN) structural framework. In our architecture, the generative DBN is used to initialize the DNN weights and the fine-tuning from the backprop is carried out sequentially layer by layer. Although DNNs are very powerful tools designed for use in both classification and recognition tasks, it is computationally abhorrent [183]. A necessary drawback of a generative architectural approach involves the use of stochastic gradient decent methods which does not scale well to high dimensionalities [242]. The backpropogation gradients increase exponentially with increasing depth and feature dimensionality [242]. Thus, for a small problem sized network with a few tens to hundreds of training data, the DNN converges remarkably to the target distribution of states (as so will other approaches). However, for a large scaled problem definition, that requires thousands or more sets of training features, DNNs will not be able to converge as accurately and efficiently to the target predictions. A portion that is attributable to this loss of efficiency can be said to have been contributed from the problem of overfitting [166], [165]. Overfitting is a mathematical phenomena which is best described as the resulting increasing diminishing returns of improvement in prediction performance, from the constant addition of more and more training feature data sets at the input [194]. Another more serious problem which Deep architectures face from non-optimized initialization points is underfitting. In this scenario, because the features used in the DBN to train the DNNs are insufficient in terms of complexity of representation, it is unable to model the weights between hidden layers of vectorized training sets effectively. This will eventually lead to

the familiar problem of vanishing and exploding (unstable) gradients [167]. Vanishing gradients occur when the differentiable chained rule of weights in between layers of training vectors approaches zero. When this happens, the weights carried over to the upper layers vanish and hence, the model breaks down completely. However, to represent a large array of dimensional complexity in a single belief architecture becomes computationally prohibitive when used to train DNNs with (especially during the fine tuning phase) [243]. Although still, generative DBNs offers many benefits like a supply of good initialization points, the efficient use of unlabeled data, it is a Bayesian probabilistic conditional that factors in correlational contributions when making a decision, it computes hidden variables efficiently even from within the deepest layers of the belief model, it is a generative distribution pre-training phase that can efficiently characterize high-order correlation properties between observables; thus, making its use in deep network architectures indispensable [163]. In order to tackle the problem of computational efficiency and learning scalability to large data sets, we have adopted the DSN model framework for our study. Central to the concept of such an architecture is the relational use of stacking to learn complex distributions from simple core belief modules, functions and classifiers. In addition, a tensorized variant provides for a proficient generalization of DSNs through high-order multivariate feature interactions. The method of approach however, remains the same between DSN and TDSN. The overall architecture of the Tensorized Deep Stacking Network (TDSN) consists of several hidden feature training primitive sets per layer; as opposed to the conventional DSN, which comprises a single hidden unit set per hidden layer. The training primitives which we have used for our TDSN model are sentiment, frequency, trust, status, influence, membership, similarity, preference and reciprocity disparities. The added benefit of generalization which this model presents means that training on the architecture in one data set (like e.g. google plus) can be re-used in another data set (like e.g. twitter, facebook, youtube, blogger, etc.).

4.4 The Generative DBN-RBM Stack

A Boltzmann Machine is architecturally defined as a stochastically coupled pair of binary units. These units contain a visible layer given as: $V \in 0, 1^D$

and a hidden layer vector: $H \in 0, 1^P$. This structural definition is given in (2.32). It can be seen that the energy terms between layers are symmetric interactions through pre-defined weights. Hence, the probability distribution over both visible and hidden layers is given by:

$$P(V, H; \theta) = \frac{\exp(-E(V, H; \theta))}{Z} \tag{4.6}$$

Where Z is the partition function given by:

$$Z = \sum_{V} \sum_{H} \exp(-E(V, H; \theta)) \tag{4.7}$$

The conditional probabilities of both visible and hidden layers within model parameter constraints are then given as:

$$P(h_j = 1|V, H_{-j}) = \sigma(\sum_{i=1}^{D} W_{ij}\nu_i + \sum_{m=1\ j}^{P} J_{jm}h_j) \tag{4.8}$$

And

$$P(\nu_i = 1|H, V_{-i}) = \sigma(\sum_{j=1}^{P} W_{ij}h_j + \sum_{K=1\ i}^{D} L_{ik}\nu_i) \tag{4.9}$$

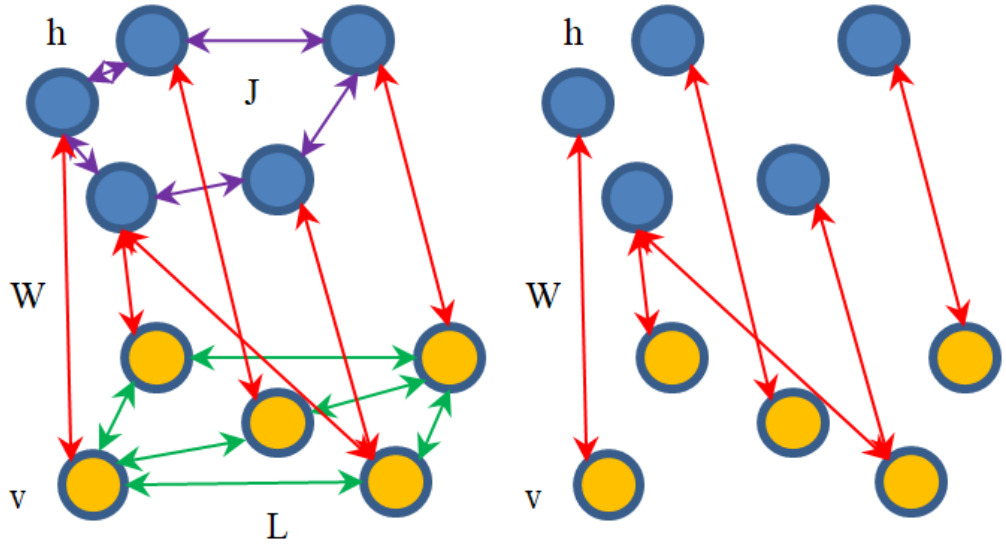
Where σ is the sigmoidal scaled activation function given in (2.35). The sigmoidal function was the original logistic function developed to emulate the activation of neurons in brain cells. However, for binary truth assignments, it has been superseded computationally, by more popular ReLU and leaky ReLU variants that has two major benefits of sparsity and a reduced likelihood of vanishing gradients of the latter over the former. A more detailed explanation and justification of our choice of activation functions is given in section 3.6 of the study. The gradient ascent in the log-likelihood can be derived from the probability distribution function as:

$$\triangle W = \alpha(E_{P_{data}}[VH^T] - E_{P_{model}}[VH^T]) \tag{4.10}$$

$$\triangle L = \alpha(E_{P_{data}}[VV^T] - E_{P_{model}}[VV^T]) \tag{4.11}$$

$$\triangle J = \alpha(E_{P_{data}}[HH^T] - E_{P_{model}}[HH^T]) \tag{4.12}$$

Where α is the learning rate, $E_{P_{data}}$ is the data dependent expectation and $E_{P_{model}}$ is the model's expectation.



a) A Classical Boltzmann Machine – The top layer represents the stochastic binary hidden vector while the bottom layer denotes the stochastic binary visible variables.

b) A Restricted Boltzmann Machine with no hidden to hidden and visible to visible intra layer connections.

Figure 4.01. Differences in architectures between Boltzmann Machines. This diagram shows key differences in node connections between layers of the boltzmann machines for traditional and restricted variants.

For a Restricted Boltzmann Machine, further constraints are placed on the model for establishing unit independence within layers by eliminating coupled interactions between them. This reduces the energy state equation equation to (2.33). While the RBM conditional probabilities reduce to:

$$P(h_j = 1|V, H_{-j}) = \sigma(\sum_{i=1}^{D} W_{ij}\nu_i + a_j) \tag{4.13}$$

And

$$P(\nu_i = 1|H, V_{-i}) = \sigma(\sum_{j=1}^{P} W_{ij}h_j + b_i) \tag{4.14}$$

Our DBN framework consists of stacking RBMs together on top of each

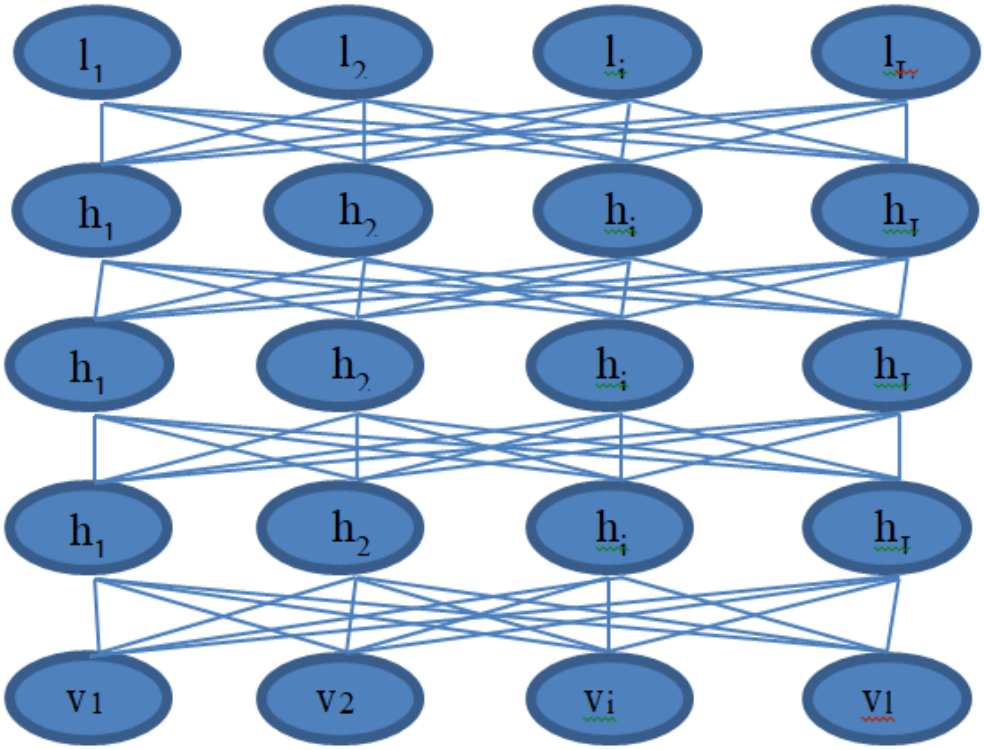


Figure 4.02. Illustration of the RFT DBN/DNN architecture framework. This diagram shows how the generative framework of the RFT model is logically structured between the layers of the deep neural stack.

other which are learned layer by layer from the bottom up. Stacking is done simply with the hidden layer of the first RBM (which are composed of activation units) being made as inputs to higher stacked layers of RBMs above it. This is illustrated in Figure 4.02.

Each layer hidden above the visible layer represents the activity vector of the trained RBMs which are then fed as visible inputs into another RBM stacked above it. The top layer is the predicted labeled layer of link feature states that can be used to determine and detect turbulence in a social relational state of flux.

4.5 The Discriminative TDSN-RNN Architecture

All deep architectures (Contrastive Divergence or per layer RBM to supervised backpropogation perceptron golden architecture) rely on a back and forth recursive process through three core stages of their learning process given in section 2.7.2. Stage 1 involves a sequential forward processing of stacked training layers from known input vectors to visible output training sets. This is also known as the forward pass. The objective function of the forward pass is the discovery of the error function between actual and trained output vectors. Once this is done, a reverse pass (also known as backpropogation) is performed in sequence from the layer by layer stack from the output to derive at the known inputs. This method chains the gradients known between training layers together to estimate the scores of the next layer below. The objective function of such a process is to learn the weights associated between training layers of the stack through a method known as a stochastic gradient descent. Finally, to wrap up this single recursive step, the weights are adjusted according to an optimization function which for obvious reasons; would want to minimize the errors (both objective and propagation) of the expectation. This three step process is repeated in a back and forth pendulum rocking fashion, until the expectation is reached.

While powerful in its approach, it is computationally prohibitive, especially when calculating stochastic gradients. Because the core approaches of the gradient descent method borrows from principles - embedded in Bayesian learning theories, they share the same computational challenges of slow convergence, saddling at learning through local minimas, and requiring extensive resources. This learning process does not scale well to increasing layers of complexity trained by large scale DBN-DNN architectures.

Therefore, a new architecture was adopted to tackle the problem of exponentially increasing computational gradients with growing complexity of deep neural layers. The idea is derived from the perceptron architecture where in 2006, Hinton discovered that if the undirected connections of hidden RBMs were individually trained using an unsupervised learning technique (Contrastive Divergence), then the whole DNN network can be later trained with human supervision using backpropogation in order to add the required fine

tuning of weights for the complex architecture more efficiently.

DSN (originally termed Deep Convex Networks or DCN) emulates Hintons discovery by composing simple modules of relational feature functions that are stacked on top of each other in order to learn more complex relations on a wider and larger scale. In this architecture, the output of each trained simple module stack is used as a hidden input feature vector to the known feed-forward neurons of another simple module stacked on top of it. A graphical representation is given in Figure 2.20.

4.6 The Hybrid RFT Fractal Architecture

We begin our subject of research with the definition of a soft kernel used to discover a markovian structure which we then encode into confabulations of fractal sub-structures. For a given set of data observables as inputs: $\chi \in X$ and outputs: $\Im \in \Xi$ we wish to loosely define a mapping such that the source space (X, α) maps onto a target space $(\Im, \omega)$. The conditional $P(\chi \vee \omega)$ assigns a probability from each source input χ to the final output space in ω. Each posterior state-space from in between input to output is generated and sampled through a random walk process. It is worth noting that markovian random walks are used to build a more generalized stochastic discovery process in our experiment. However for larger datasets, any one of the more sophisticated markovian sampling methods (e.g. Gibbs, Monte carlo, Metropolis-Hastings, Hamiltonian, etc.) can be used as drop-in replacements. An indicator function which we have chosen to describe the state transition rule is:

$$\Theta_{t+1} = min \begin{cases} 0 \\ \amalg_{c=1}^{n} \frac{\delta E_{t+1}^{c}}{\delta \chi_{t}^{c}} \end{cases} \tag{4.15}$$

Where δE_{t+1}^{c} is the error change from one hidden feature activity state $h_t \in H$ onto higher posterior confabulations. The objective function at each transition seeks to minimize error gradients to eliminate problems associated with exploding and vanishing gradients during backpropagation. This can be caused by an excessive generation of layered confabulations which leads to

unnecessary increments in depth from the markovian ANN discovery mechanism. For a general finite state space markovian process, the markov kernel is thus defined as:

$$Kern(M) = \begin{cases} p : X \times \omega \to [0,1] \\ p(\chi|\omega) = \oint_{\omega} q(\chi, \Im)\nu(\delta\Im) \end{cases} \tag{4.16}$$

Once a unique markovian neural network has been discovered, a Single Layer Convolutional Perceptrion (SLCP) is proposed as a baseline structure to learn the fractal sub-network from pre-existing posterior confabulations. The SLCP baseline structure (Figure 4.03) changes as discovered knowledge is progressively encoded during the learning process. Although for simplicity we have used the novel SLCP architecture as our baseline schema; in reality however, any one baseline model can be used to learn a morphing transposition into a fractal signature structure. In essence, methods like Progressive Neural Networks (PNNs) where activation links of neighboring DNN stacks are learned laterally across hidden layers [244] or the wide use of summarizing information from ensemble methods like distillation [204] are relevant alternatives.

From the viewpoint of a DNN architecture, we can define an input to output transition broadly as:

$$Y_n = KX_n + B \tag{4.17}$$

Where n corresponds to the number of layers in the stack, Y_n is the expected output, X_n is the data at the input and B is the network bias. K is a unique signature of transpositions of hidden activities from source to destination data spaces which we wish to capture and encode into RFT. For an N-deep neural network, K can be expressed as a chained state of lower to upper activation weights:

$$\prod_{n=1}^{N} U_n^T \sigma W_n^T \tag{4.18}$$

Where T is the target tensor, U^T are the upper layer target vector weights and W^T are the lower layer weights. In our model, σ (sigmoid logistic func-

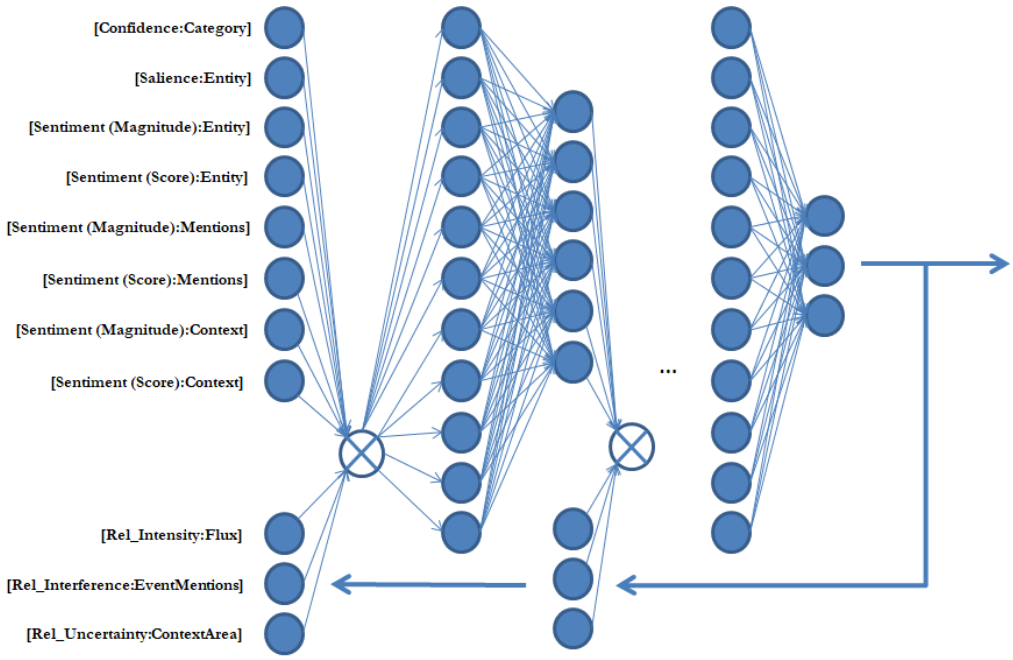


Figure 4.04. The RFT architecture design. This diagram shows how the RFT system is designed from data inputs, pre-processing, sentiment analysis and relational turbulence extraction.

tion) was arbitrarily chosen as the default activation for each neuron. Since posterior hidden activities $h_n \in H$ are known, then an error derivative which converges to zero gives:

$$U = (HH^T)^{-1}HT^T \tag{4.19}$$

Which essentially states that upper layered weights maintain a canonical property of identical symmetry and reversibility about each hidden layer activity H and the target T. The hidden layer encoding technique was built using the following optimal maximum entropy constraint:

$$min_{P(\psi|\phi)} - H(P) = \sum_{\psi_e,\psi_z} \bar{P}(\phi_z)P(\psi_e|\phi_z)\log P(\psi_e|\phi_z) \tag{4.20}$$

Where ψ_e are the feature transposed structure states and ϕ_z are the given posterior hidden feature weights. The transposed structure state ψ is given simply by the mathematical relation as:

$$\sum_{e=1}^{n} \psi_e = \sum_{z=1}^{i} \Theta_z\phi_z \oplus k_z\psi_z + \rho_z \tag{4.21}$$

Where k_z is the morphing constraint on the trans-positioned structural states ψ_z in z posteriors and ρ_z is the normally distributed transpositioned errors as: ρ_z $N(0, Q_z)$. Such that Q_z is the error covariance matrix. The estimation of $P(\psi|\phi)$ that minimizes $H(P)$ is done using the Lagrange parameters ς. The convex solution is then estimated as:

$$P_\varsigma(\psi|\phi) = \frac{1}{D_\varsigma(z)} \exp(\sum_{n=1}^{F} \varsigma_n f_n(z, e)) \tag{4.22}$$

Where n denotes the number of features measured in a dataset. Once feature entropies have been encoded into the fractal subnetwork, this structure is then used to generate depths for highly sophisticated model representations. This is done through de-quantization of the entropy decoded (expanded) model as:

$$G^{k'}(\psi|\phi) = G^k(\psi|\phi) \times k(\psi, \phi) \tag{4.23}$$

Where $G^{k'}(\psi, \phi)$ is the de-quantized graphical representation of the confabulations to the sophistication levels of a feature tensor from the previously encoded fractal structure $G^k(\psi|\phi)$. A safe stopping condition is triggered when error gradients approaches zero for expectations to converge. The theoretically desired condition is given as:

$$E(Y) = P(Y|X) \forall \frac{\delta E_n^c}{\delta \chi_n^c} \rightarrow 0 \tag{4.24}$$

Where $\frac{\delta E_n^c}{\delta \chi_n^c}$ is the error gradient of each confabulation at every epoch.

4.7 The Forward Pass and Loss Function Discovery

In a Fractal Neural Network (FNN) built of L layers where layer 1 and layer L represent the input and output layers respectively, we wish to determine the dimensions of real output to input as: $R^{n_1} \rightarrow R^{n_L}$ for some layer output R^{n_l} where $l \in L$. The forward pass discovers the weighted outputs from inputs of individual confabulations. The weight of a given layer l is written as: $W^{[l]} \in R^{n_{l-1}}$. For a given input $\chi \in R^{n_1}$, the activity of the forward pass

can be mathematically summarized as:

$$a^{[1]} = \chi \in R^{n_l} \tag{4.25}$$

at the input. Correspondingly, at upper layer l of the forward pass, the activity is mathematically defined as:

$$a^{[l]} = \sigma(W^{[l]}a^{[l=1]} + b^{[l]}) \in R^{n_l} \tag{4.26}$$

Where $b^{[l]}$ is the weight bias at the l layer of the FNN architecture. Finally, at the output layer L of the forward pass, the neuron activity is given as:

$$a^{[L]} = \sigma(W^{[L]}a^{[L=1]} + b^{[L]}) \in \Re^{n_L} \tag{4.27}$$

For N batches of training data, the generalized loss function is mathematically expressed as:

$$C = min\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}|y(x^i) - a^{[L]}(x^i)| \tag{4.28}$$

Where C is the cost function of all weights and biases taken into consideration.

4.8 Backprop and Fine Tuning

Due to the dynamic stacking nature of the FNN model used in our study, the backpropagation techniques which we use in our model are the first order Stochastic Gradient Descent (SGD) with momentum and the second order sub-sampled Trust Regions (TR) and Adaptive Regularization with Cubics (ARC) methods [226]. First order gradient descent methods are highly versatile and computationally efficient in comparison to second order gradient descent methods. For our learning model, when the fractal stack is small, first order gradient descent methods perform remarkably well at convergence and computational efficiency. Given a sequence of vectors R^S intuitively, we wish to minimize the cost function C by introducing a perturbation at the upper layer $\triangle\theta$ to the current upper layer vector p such that $min\triangledown C(\theta)^T\triangle\theta$. For that to occur, a small stepsize α is required to propagate the update rule

from the top layers down:

$$\theta\prime \rightarrow \theta - (\varrho\nu + \alpha \triangledown C(\theta)) \tag{4.29}$$

The stepsize α is the learning rate. Where ϱ is the memory constant and ν is the velocity vector having represented with the same cardinality as θ. With successive updates, the cost improvement becomes:

$$\triangledown C(\theta) = \frac{1}{N} \sum_{i=1}^{N} \triangledown C_{X^i}(\theta\prime) \tag{4.30}$$

Although simple and powerful in their approach, first order SGD with momentum isnt without its weaknesses. Its main drawback is that they only consider first order gradients which include a wide range of flaws. These flaws include relatively slow convergence rates, ultra-sensitivity to hyper parameters, ill-conditioning and bad initialization points, flat gradients and gradient saddle points which lead to high generalization errors and vanishing gradients [245].

Subsampled TR and ARC methods belong to the Newtonian type of approaches which are designed to deliver performance of deep learning techniques under a non-convex solution manifold. They achieve this by estimating curvature information of the manifold in Hessian representations [226]. Because they are able to achieve convergence to second-order critically, TR and ARC approaches address several shortcomings of the first-order SGD.

TR and ARC approaches are able to compete computationally to SGD methods especially when depth of the FNN grows in complexity to adequately fit into representing the sophistication of the RTM in question. Secondly, they are robust enough to sensitive hyper-parameter tuning mechanisms. Thirdly, they are capable of avoiding gradient saddle regions and thereby adapt towards converging at low generalization errors. Fourthly, the sub-sampling approach provides higher bandwidth and therefore tolerance for error-correction mechanisms. Finally, TR and ARC approaches provide superior performance measures against first order SGD with momentum and second order Gauss-Newton (GN) and limited-memory BFGS (L-BFGS) methods.

The generalized loss function can be re-written mathematically as a non-convex finite sum minimization in the form:

$$min_{x\in\Re^d} C(x) \hat{=} \frac{1}{N}\sum_{i=1}^{N} C_i(x) \tag{4.31}$$

Where

$$C_i(x) = \frac{1}{2}|y(x^i - a^{[L]}(x^i))| \tag{4.32}$$

As previously mentioned, C is the error cost function that corresponds to empirical risk of the predicted tensors at the output. The application of an exact Hessian curvature form to solve a second order convergence is computationally infeasible because of the infinite number of possible curvature solutions in a given manifold. In an ultra-deep network where $N, d >> 1$, evaluations of the Hessian and gradient increases linearly in N. Instead, from a given sampling distribution $P_{i=1}^{N}$ over N batches of training data; considering a sub-sampled distribution for these posteriors in a manifold yields a likelihood profile given mathematically as:

$$H(x) \hat{=} \frac{1}{N|S|}\sum_{j\in S}\frac{1}{P_j}\nabla^2 C_j(x) \tag{4.33}$$

Which essentially estimates the curvature profile $\nabla^2 C_j(x)$ of the cost solution manifold through a likelihod measure $\frac{1}{P_j}$ at where probability densities are highest in the posterior distributions.

The Trust Region (TR) sub-problem can then be represented as:

$$S_t \approx argmin_{|S|\leq\triangle_t} m_t(S) \hat{=} \langle \nabla C(x_t), s\rangle + \frac{1}{2}\langle s, H_t s\rangle \tag{4.34}$$

Respectively, the Cubic Regularization (ARC) sub-problem can be expressed mathematically as:

$$S_t \approx argmin_{|S|\in\Re^d} m_t(S) \hat{=} \langle \nabla C(x_t), s\rangle + \frac{1}{2}\langle s, H_t s\rangle + \frac{\sigma_t}{3}|S|^3 \tag{4.35}$$

Where $|S|$ in both equations represent TR and ARC regularization terms respectively.

The backpropagation chain rule pathway gives:

$$\frac{\partial C}{\partial W_{jk}^{[l]}} = \delta_j^{[l]} a_k^{[l-1]} \tag{4.36}$$

Where $\frac{\partial C}{\partial W_{jk}^{[l]}}$ is the back propagated weight change from the estimated cost corrected gradient descent TR and ARC optimization. $\delta_j^{[l]}$ is the j^{th} error tensor at the l layer and $a_k^{[l-1]}$ is the k^{th} neuron activity at the $l-1$ layer.

4.9 Activation and Anti-Aliasing

Our model uses the ReLU activation function. ReLU activation functions help by reducing the likelihood of a vanishing gradient. This occurs when input samples increases above zero ($a > 0$). In this situation, ReLU functions define activations to increase linearly with a constant positive gradient. Furthermore, when input samples fall below zero $a \leq 0$ ReLU immediately drops the gradient to zero, thereby making the computation of weights intractable and deactivates the neuron [162]. In contrast however, a sigmoidal function represents the activation with decreasing positive gradients as the positive input samples increases. This means that for sigmoidal activation functions, learning slows down as the number of positive training samples grows larger (as will be the case in highly stacked deep networks) [163]. ReLU functions therefore, provide for faster learning mechanisms.

The other benefit of ReLUs is sparsity. ReLUs are capable of adding to sparsity of the distribution when negative input samples exist ($a \leq 0$). The more such units exist in a layer, the more sparse the resulting representation will be. By comparison, because Sigmoids are a negative inverse reflection of its positive real valued counterpart, dense representations of some non-zero valued activations tend to persist throughout the learning process. This makes for bad training samples where sparsity of activations are required to effectively learn a pattern for tasks like recognition, classification, etc. [170].

4.10 Experiments and Results

4.10.1 Experimental Data

The experiments were conducted on three datasets using three different algorithms. The datasets are: Twitter, Google and Enron emails. The Stanford Twitter Sentiment Corpus contains APIs[1] for classifying raw tweets that allows us to integrate their classifiers into our deep learning model. Their plug-in module uses an ensemble of different learning classifiers and feature extractors to deliver the best outputs with different combinations of classifiers and feature extractors. In addition to the sentiment results obtained from their model, we cross validated the output against googles NLP API[2] to replicate the most accurate sentiment scores and magnitudes of context spaces and mentions.

The Google dataset was obtained from the repositories of common crawl and was sentilyzed from the stripped down WET file contents. The dataset which was used in this experiment was extracted from the April 2014 crawl data. Lastly, the Enron email dataset was obtained from the David Newman website, hosted on the UCI Machine Learning Repository[3]. The entire repository of email contents were extracted and sentilyzed using googles NLP model to provide the inputs we require of our training model.

4.10.2 Experimental Design

Figure 4.04 describes the inputs into our model. Specifically, the RFT model accepts as inputs, the confidence of the detected category in every social transaction, the Salience of all detected entities in the transaction, the sentiment scores and magnitudes of entities, mentions and drifting contexts. These eight relational features form the key independent input into our RFT fractal neural network (FNN) model. Additionally, the outputs (Relational Intensity γ_{rl}, Relational Interference ϑ_{rl} and Relational Uncertainty φ_{rl}) which represent turbulence are fed back into the model as recurrent in-

[1] http://help.sentiment140.com/api
[2] https://cloud.google.com/natural-language/
[3] https://archive.ics.uci.edu/ml/datasets/bag+of+words

puts into the neural network to act as memory retention for the relational turbulence profiles of previous transaction/s, and as good influential initialization points for new training sequences of extracted sentiments in later social transactions.

Relational Turbulence was calculated from conditional posteriors of γ_{rl}, ϑ_{rl} and φ_{rl} as the mathematical relation of:

$$P(\tau_{rl}) = \sum_{i=1}^{n} \frac{P(\gamma_i|\theta_i)P(\vartheta_i|\varphi_i)P(\varphi_i|\gamma_i)}{N_i P(\gamma_i)P(\vartheta_i)P(\varphi_i)} \tag{4.37}$$

The inputs were tested across three deep architecture models and the learning results were compared using both Kendall and Spearman correlation tests to measure both strength of dependence and degree of association between input independent variables and output turbulence metrics. In addition, the different deep learning approaches were cross validated using k-fold cross validation techniques.

4.10.3 Experimental Findings

The tests were run across the Single Layer Perceptron (SLP), a 45-layer Deep Convolutional Network (DCN) and a dynamically stacked Fractal Neural Network (FNN). The results are shown in Table 4.01 - 4.11 and Figure 4.05 - 4.30:

4.10.4 Performance Measurements

The Kendall (w coefficient) and Spearman (rho coefficient) tests were conducted on the results obtained from the testing procedures.

Specifically, the Kendall (tau-b coefficient) was used to measure the strength of associations between predicted and expected outputs of the learning mod-

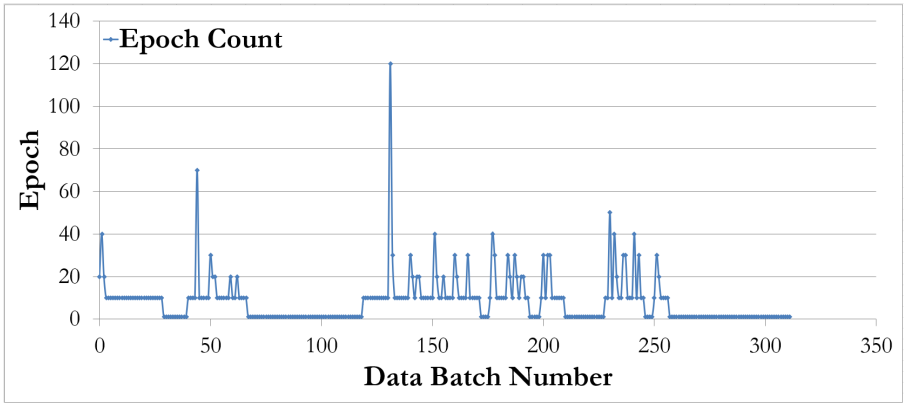


Figure 4.05. Graph of Twitter learning rate convergence for the SLP model

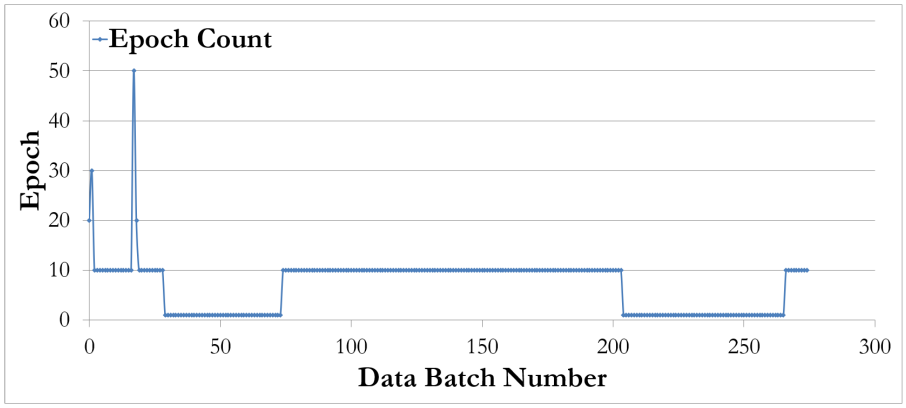


Figure 4.06. Graph of Twitter learning rate convergence for the DCN model

els. The Kendall (tau-b) coefficient is given as:

$$\tau_b = \frac{N_c - N_d}{\sqrt{(N_0 - N_x)(N_0 - N_y)}} \tag{4.38}$$

Where,

$$N_0 = \frac{N(N-1)}{2} \tag{4.39}$$

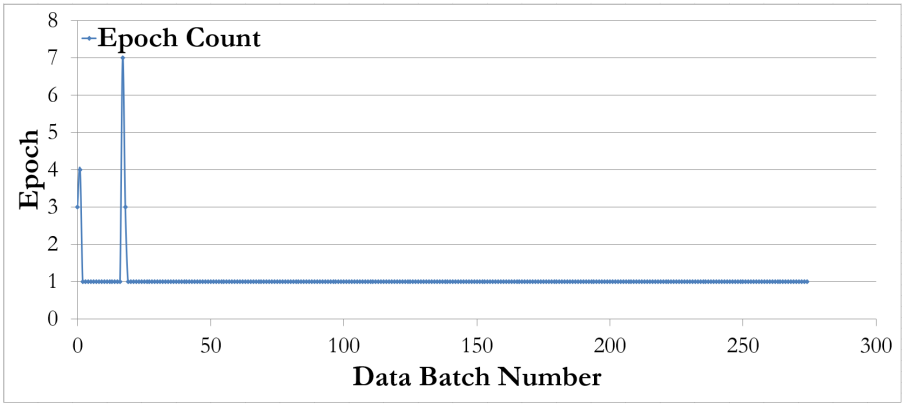


Figure 4.07. Graph of Twitter learning rate convergence for the RFT model

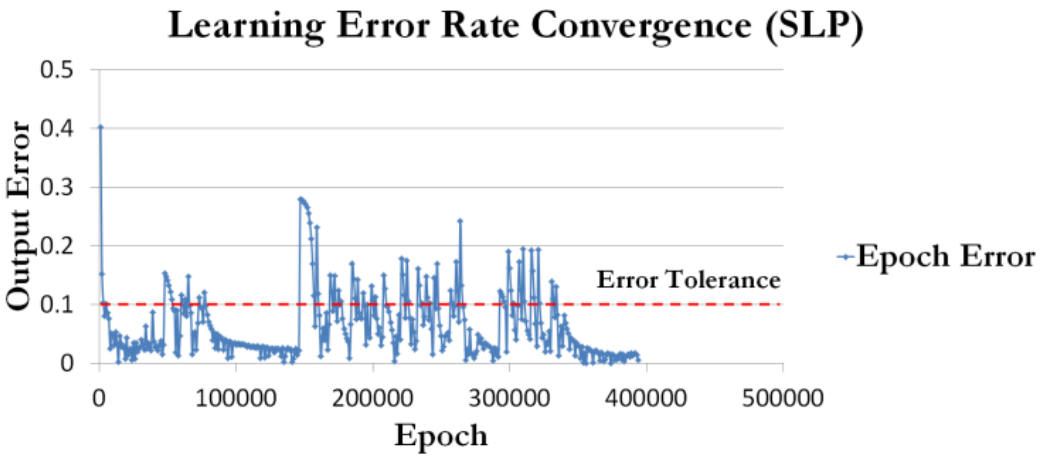


Figure 4.08. Graph of Twitter error rate convergence for the SLP model

And,

$$N_x = \sum_i \frac{u_i(u_i - 1)}{2} \tag{4.40}$$

And,

$$N_y = \sum_j \frac{v_j(v_j - 1)}{2} \tag{4.41}$$

Where N_c is the number of concordant paris, N_d is the number of discordant pairs, u_i is the number of tied values in the i^{th} group of ties for the first quantity and v_j is the number of tied values in the j^{th} group of ties for the second quantity.

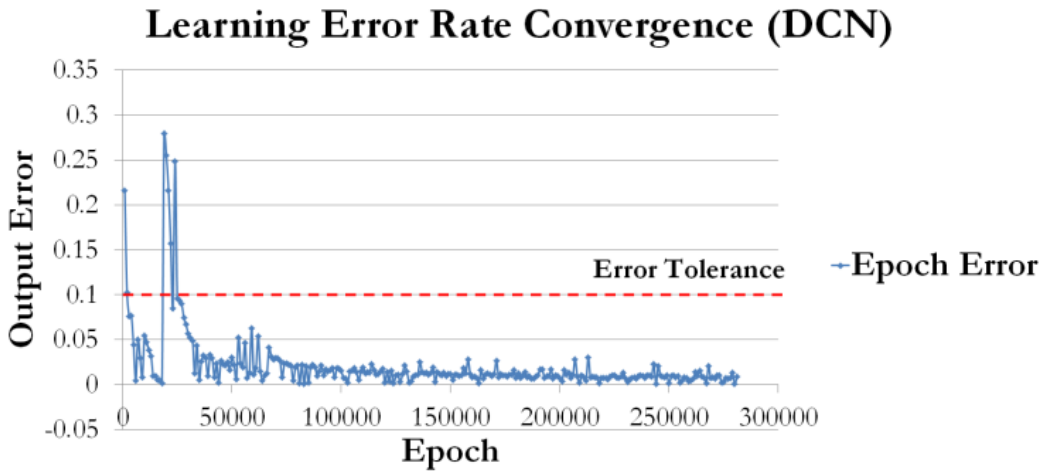


Figure 4.09. Graph of Twitter error rate convergence for the DCN model

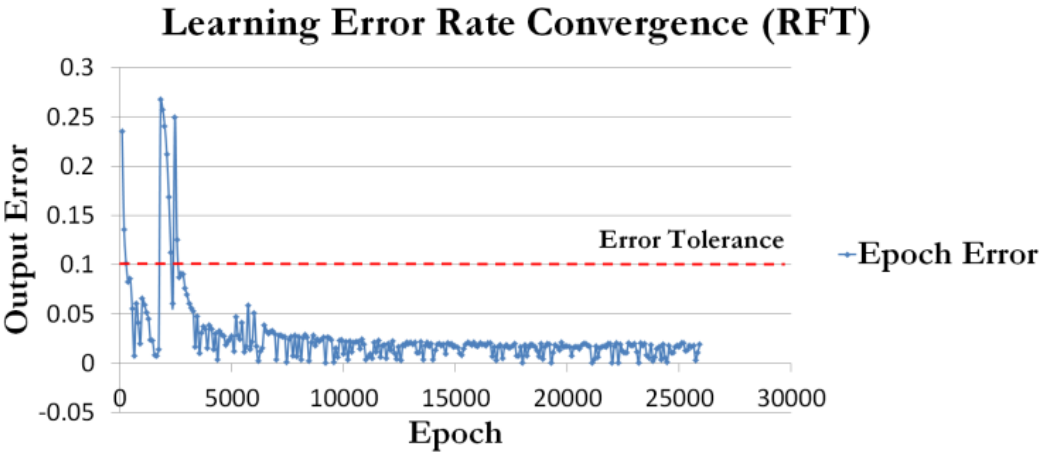


Figure 4.10. Graph of Twitter error rate convergence for the RFT model

The Spearman (rho coefficient) was used to measure the monotonic relationship between the independent variables (Category confidence $\mathfrak{C}_i$, Entity Sailence $\mathcal{J}_i$, Entity sentiments - magnitude and scores ($\Im_i, \beth_i$), Mention sentiments -magnitude and scores ($\mathcal{L}_i, \lambda_i$), Context sentiments - magnitude and scores ($\mathcal{O}_i, \top_i$)) and the dependent variables (Relational Intensity γ_{rl}, Relational Interference ϑ_{rl} and Relational Uncertainty φ_{rl}). Essentially, the relationship of measure is calculated as:

$$\Gamma_S = 1 - \frac{6 \sum D_i^2}{N(N^2 - 1)} \tag{4.42}$$

Where $D_i = rank(X_i) - rank(Y_i)$ is the difference in ranks between the observed independent variable X_i and dependent variable Y_i and N is the

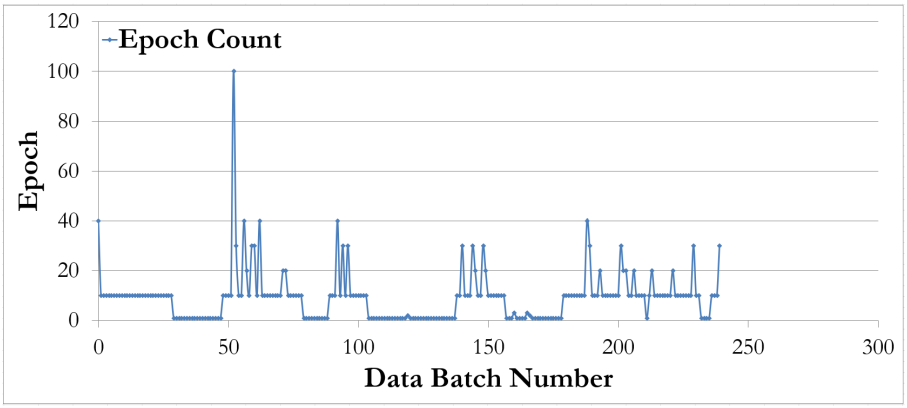


Figure 4.11. Graph of Google learning rate convergence for the SLP model

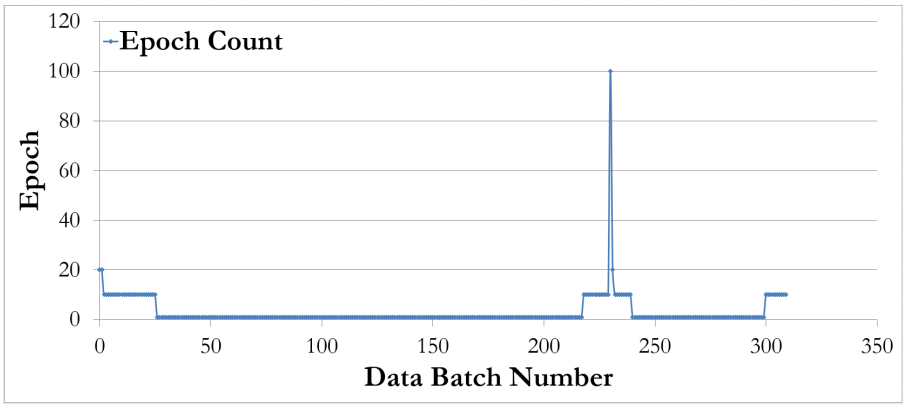


Figure 4.12. Graph of Google learning rate convergence for the DCN model

number of predictions to input data sets for all three sources.

4.10.5 Testing Results

Finally, during the experimentation, the full datasets obtained from the different sources (twitter, google and enron) were partitioned into k-subsamples. One of the subsamples was retained as the validation set for each run and

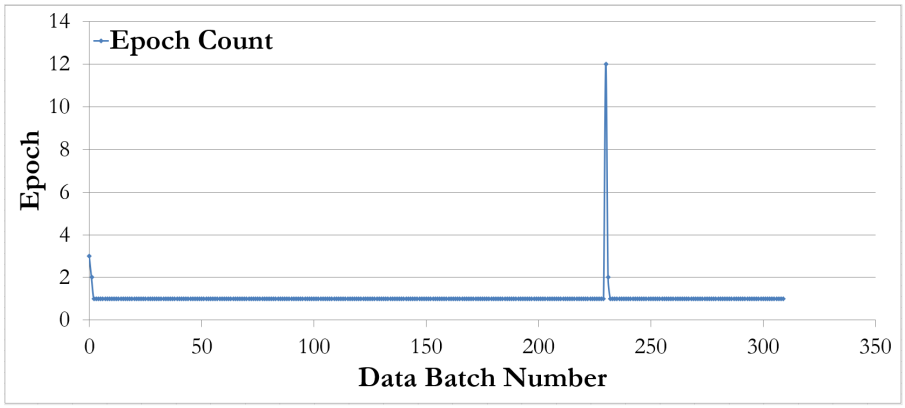


Figure 4.13. Graph of Google learning rate convergence for the RFT model

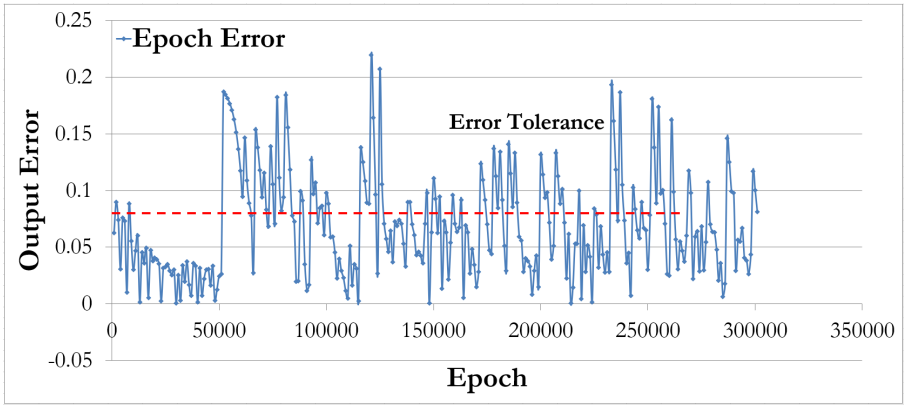


Figure 4.14. Graph of Google error rate convergence for the SLP model

the validation set was chosen in a round robin fashion for subsequent experimentation runs. A noteworthy point of mention is that K fold cross validation is used in our experimentation design to obtain a good estimate of the prediction generalization. This testing technique does not scale well to measurements of model precision. How accurately a learning model is able to predict an expected output is based on the Kendall (tau-b coefficient) results. K-fold validation was performed over all deep learning models across the Mean Absolute Percentage Error (MAPE) measurement of each

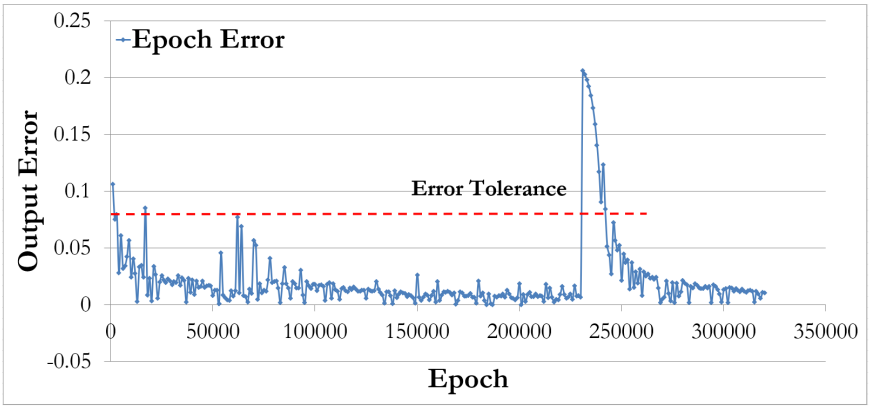


Figure 4.15. Graph of Google error rate convergence for the DCN model

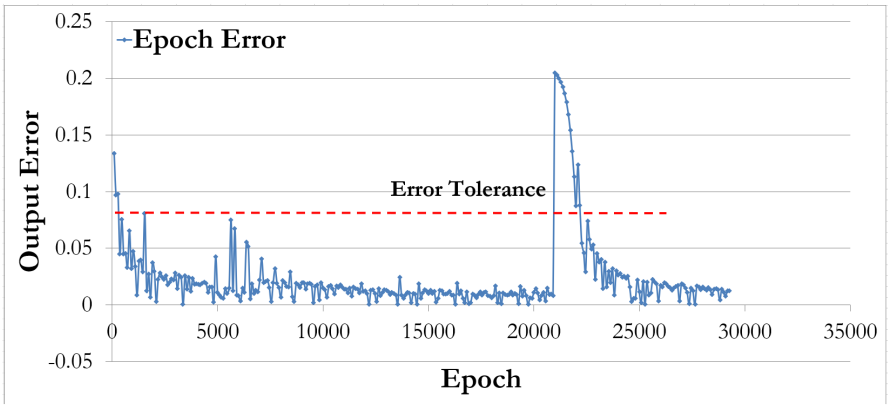


Figure 4.16. Graph of Google error rate convergence for the RFT model

run. Mathematically, MAPE can be expressed as:

$$\delta_{MAPE} = \frac{1}{N}\sum_{i=1}^{N} \left| \frac{E_i(x) - Y_i(t)}{E_i(x)} \right| \tag{4.43}$$

Where $E_i(x)$ is the expectation at the output of data input set i and $Y_i(t)$ is the corresponding prediction over N total subsamples. The tabulation of the K-fold cross validation used in our experimentation is given in Table 4.11.

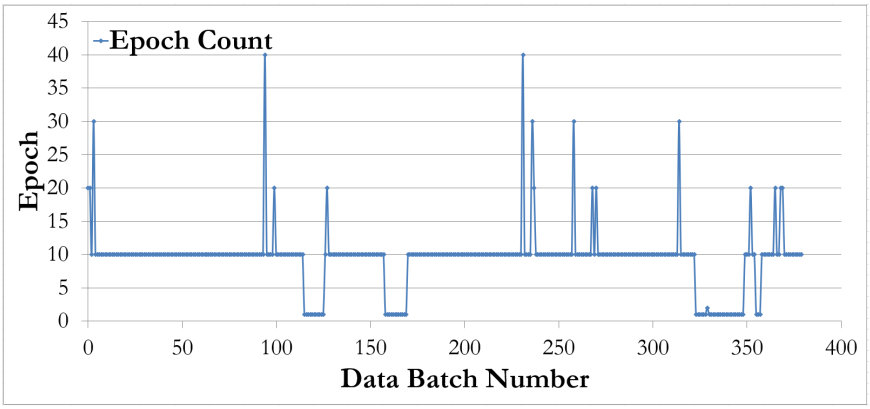


Figure 4.17. Graph of Enron email dataset learning rate convergence for the SLP model

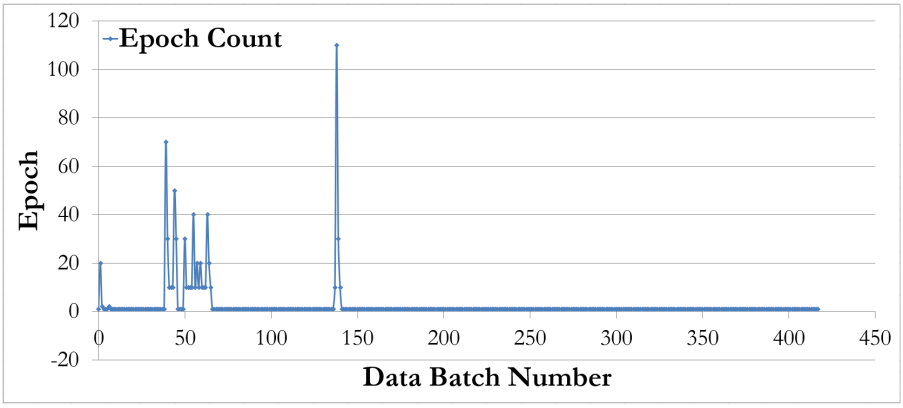


Figure 4.18. Graph of Enron email dataset learning rate convergence for the DCN model

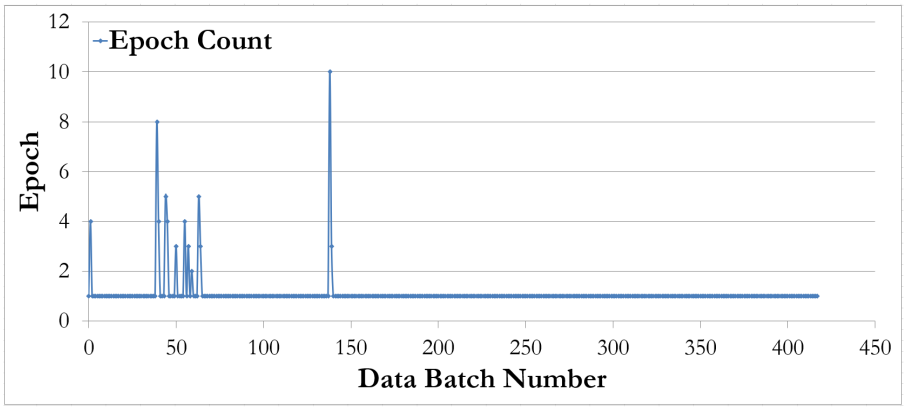


Figure 4.19. Graph of Enron email dataset learning rate convergence for the RFT model

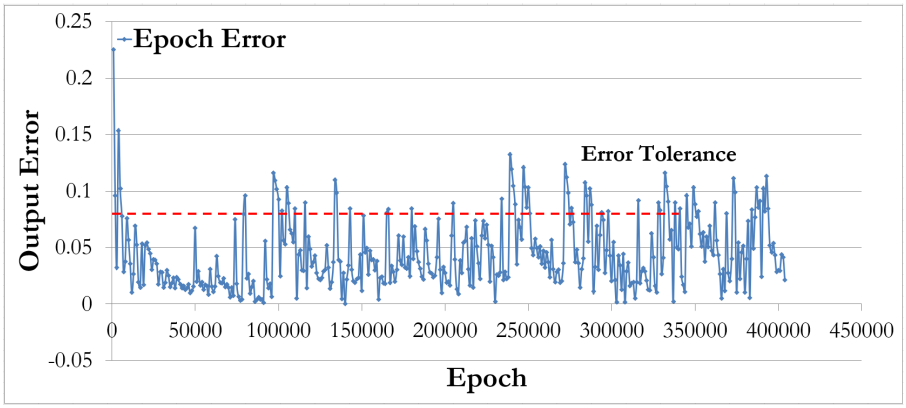


Figure 4.20. Graph of Enron email dataset error rate convergence for the SLP model

4.11 Analysis and Discussion

As can be seen from the graphs, SLP models consistently underperforms in ranking where prediction accuracy is concerned, the Kendall (tau-b coefficient) test shows a lower (positive) correlation between expected and predicted outputs across the test data set for SLP models and much higher (positive) association for both DCN and RFT. Additionally, from the results

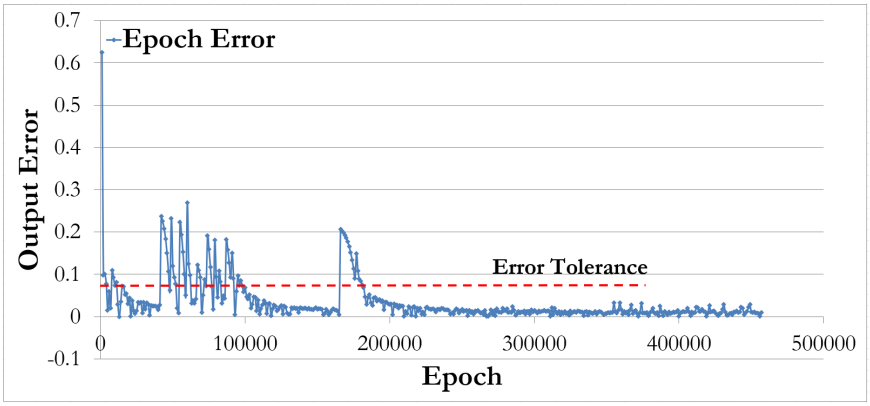


Figure 4.21. Graph of Enron email dataset error rate convergence for the DCN model

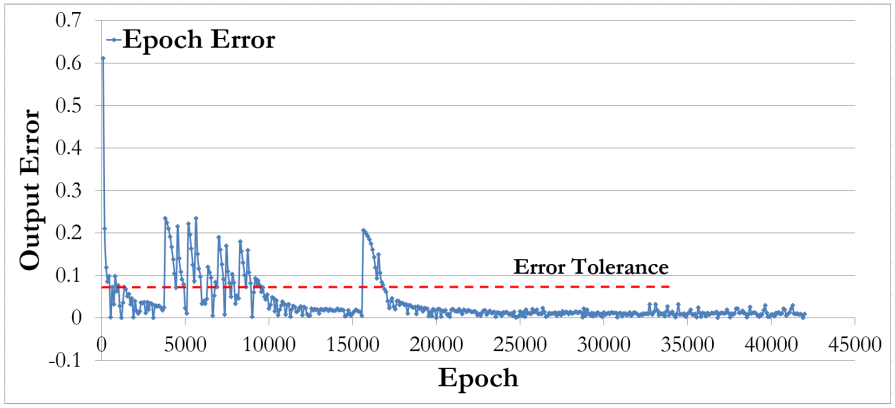


Figure 4.22. Graph of Enron email dataset error rate convergence for the RFT model

of the Spearman (rho coefficient) test done on the independent variables (Category confidence $\mathfrak{C}_i$, Entity Sailence $\mathcal{J}_i$, Entity sentiments - magnitude and scores ($\Im_i$, $\beth_i$), Mention sentiments -magnitude and scores ($\mathcal{L}_i$, λ_i), Context sentiments - magnitude and scores ($\mathcal{O}_i$, $\top_i$)) and the dependent variables (Relational Intensity γ_{rl}, Relational Interference ϑ_{rl} and Relational Uncertainty φ_{rl}), it can be seen from Tables 4.01 - 4.10 that the spearman coefficient

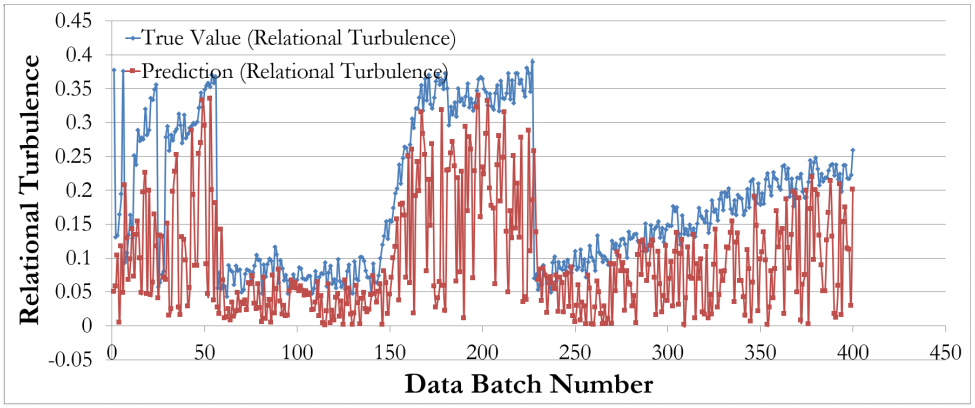


Figure 4.23. Graph of True and Predicted SLP relational turbulence values for the Twitter dataset

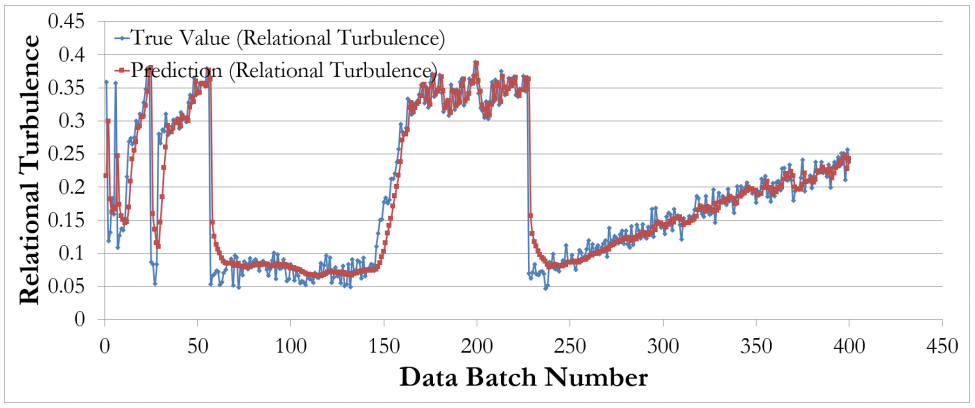


Figure 4.24. Graph of True and Predicted DCN relational turbulence values for the Twitter dataset

indicates strongly positive monotonic correlations between turbulence measures (γ_{rl}, ϑ_{rl} and φ_{rl}) and sentiment scores $[(\Im_i, \beth_i), (\mathcal{L}_i, \lambda_i), (\mathcal{O}_i, \top_i)]$ and moderately positive correlations between the same turbulence measures (γ_{rl}, ϑ_{rl} and φ_{rl}) to both category confidence and entity salience $(\mathfrak{C}_i, \mathcal{J}_i)$.

Additionally, from Tables 4.01 - 4.10, relational flux intensity, interference

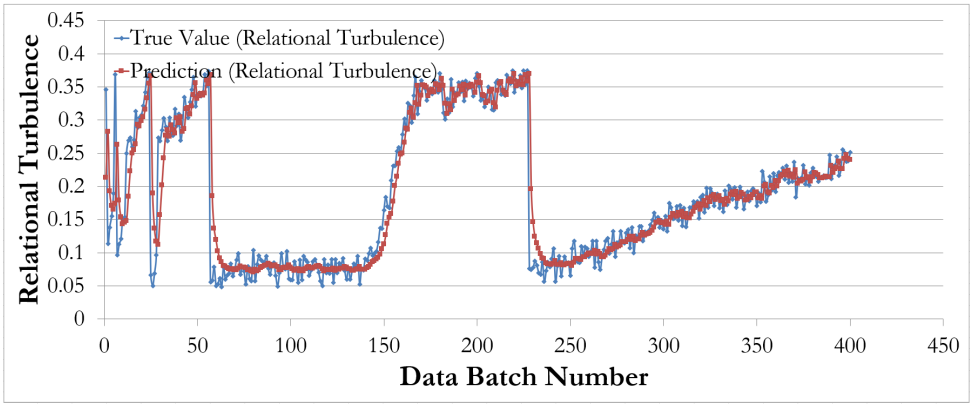


Figure 4.25. Graph of True and Predicted RFT relational turbulence values for the Twitter dataset

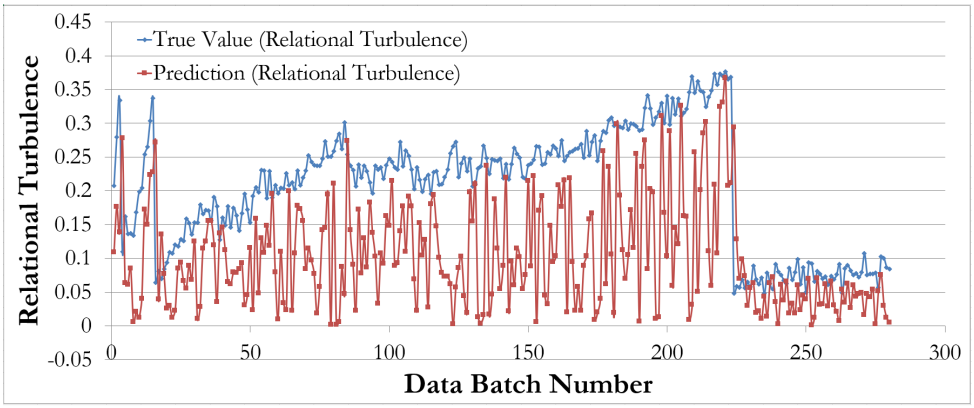


Figure 4.26. Graph of True and Predicted SLP relational turbulence values for the Google dataset

and uncertainty correlates quite strongly to category confidence in specifically directed communications. This is observed in Enrons email datasets as opposed to Twitter and Google results. Additionally, it can also be readily appreciated that the relational interference tends to correlate fairly well to entity salience scores. This can be observed from high spearman coefficients in both Twitter and Enron datasets and moderate spearman coefficients in

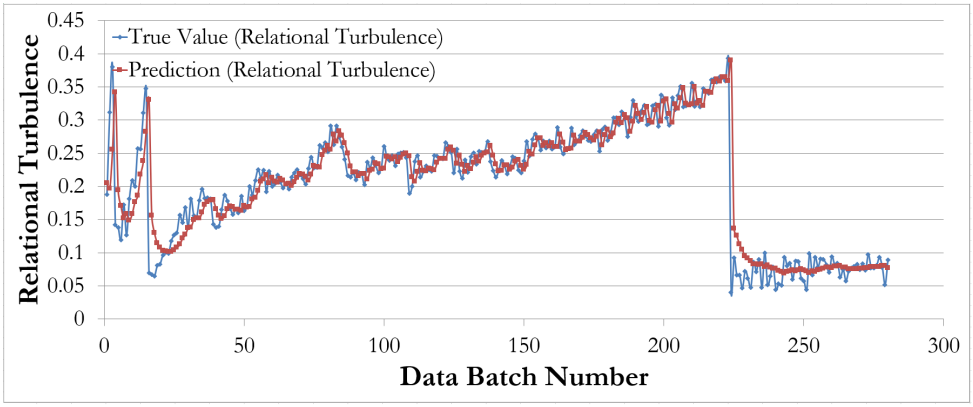


Figure 4.27. Graph of True and Predicted DCN relational turbulence values for the Google dataset

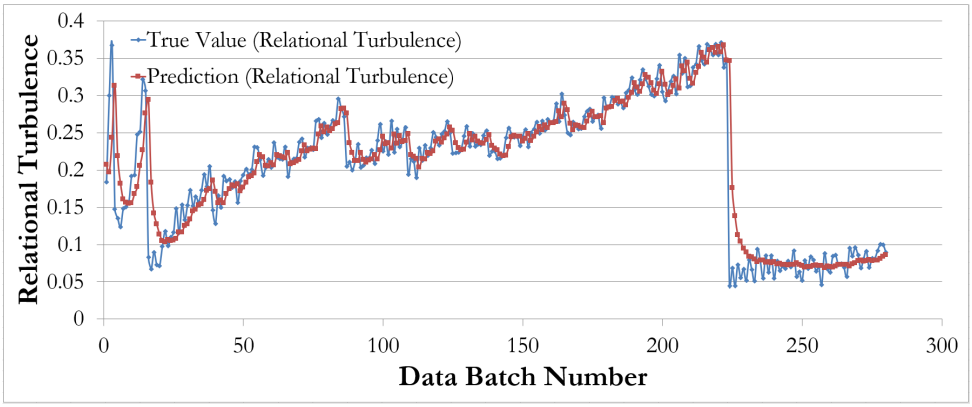


Figure 4.28. Graph of True and Predicted RFT relational turbulence values for the Google dataset

the Google dataset. This means that an actor with a higher social status of influence may more readily interfere with other relationships which are stable within a given social structure especially if their expressed sentiments go against those generally expressed in context of the transaction/s.

Generally however, it can be observed that intensity, interference and un-

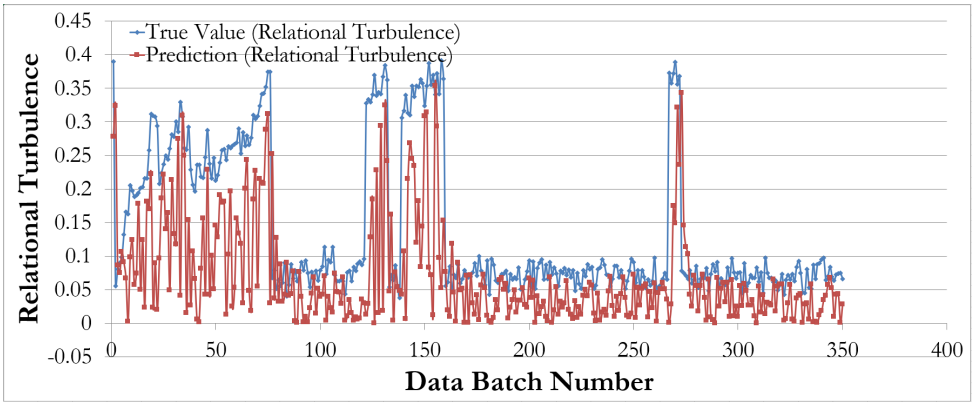


Figure 4.29. Graph of True and Predicted SLP relational turbulence values for the Enron email dataset

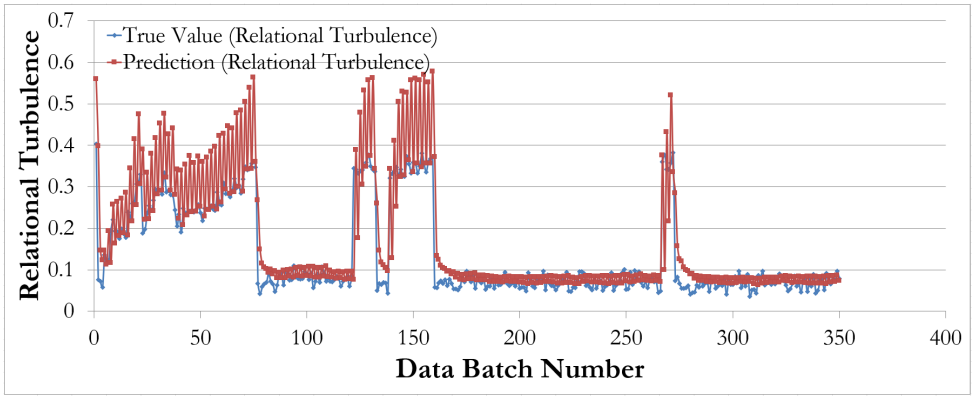


Figure 4.30. Graph of True and Predicted DCN relational turbulence values for the Enron email dataset

certainty correlates very well to expressed sentiments over entities, mentions, and (fairly well) over contexts. However, an interesting observation made from the distribution of the results is that while entity and mention sentiments are (strongly) positively correlated to the tenets of relational turbulence (i.e. higher sentiment scores expressed in these classifier manifolds are more likely to evoke a relational state altering event), context sentiments

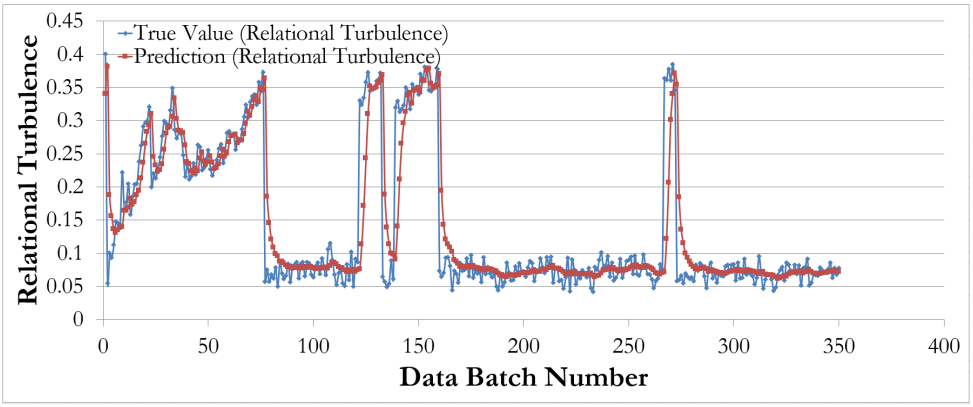


Figure 4.31. Graph of True and Predicted RFT relational turbulence values for the Enron email dataset

are (mediocrely) negatively correlated instead. It can also be observed that this negative correlation of contexts to turbulence is weaker in both Twitter and Enron (where communications are both specifically directed and / or semi-directed at social individuals) and stronger in Google datasets (where communications are non-specific and loosely directed at certain social groups or communities). Intuitively, a very realistic proposition can infer that topic contexts dont really matter within very specific and self-contained social transactions with low duplicities (a 1:1 or 1:n social transactions as in the case of Enron emails and Twitter tweets). Instead, the expressed sentiments about the details within the message become more important. Whereas however, if those same contexts were to be staged over wide ranging and broad social interactions with high duplicities (n:n social transactions as in the case of google web page datasets), then relational turbulence is inversely dependent on the expressed contextual sentiments (i.e. contexts with lower sentiment scores where opinions are largely contradictory and / or controversial, are more likely to evoke wide-spread relational state altering events).

4.12 Conclusion

In conclusion, we have shown that RFT is capable of predicting relational turbulence profiles between actors within a given OSN acquired from anytime data. Furthermore, the novel FNN model which we have developed is able to rapidly scale and adaptively represent relational complexities of anytime sequenced data within a live online social scene. Our results show superior accuracies and performance of the FNN model in comparison well known baseline models like the Single Layer Perceptron (SLP) and the Multi-Layer Perceptron (MLP) designs. We have demonstrated the feasibility of our learning model through the implementation on three large scale networks: Twitter, Google Plus and Enron emails. An important milestone in our research is that we have proven that dynamic communication patterns (derived from relational features exchanged) between actors within an OSN correlate positively to evolving relational states within a mixed eventful context model of a given social community. Progress has been made in Event Prediction from approaches covering various perspectives. However, several key important questions still remain. All of them converge to the representational accuracy of dynamic and evolving social structure nestled within an environment of constant social shocks. Our study uncovers three pivotal long-term objectives from a relational perspective. Firstly, relational features can be used to strengthen medical, cyber security and social applications where the constant challenges between detection, recommendation, prediction, data utility and privacy are being continually addressed. Secondly, in fintech applications, relational predicates (e.g. turbulence) are determinants to market movements - closely modeled after a system of constant shocks. Thirdly, in artificial intelligence applications like computer cognition and robotics, learning relational features between social actors enables machines to recognize and evolve. Deep learning relational graph models appear to have considerable potential, especially in the fast growing area of social networks.

Correlation Matrix

		True Value (SLP) - Enron	Prediction (SLP) - Enron
True Value (SLP) - Enron	Kendall's Tau B	—	0.300 ***
	p-value	—	< .001
Prediction (SLP) - Enron	Kendall's Tau B		—
	p-value		—

		True Value (DCN) - Enron	Prediction (DCN) - Enron
True Value (DCN) - Enron	Kendall's Tau B	—	0.456 ***
	p-value	—	< .001
Prediction (DCN) - Enron	Kendall's Tau B		—
	p-value		—

		True Value (RFT) - Enron	Prediction (RFT) - Enron
True Value (RFT) - Enron	Kendall's Tau B	—	0.445 ***
	p-value	—	< .001
Prediction (RFT) - Enron	Kendall's Tau B		—
	p-value		—

		True Value (SLP) - Google	Prediction (SLP) - Google
True Value (SLP) - Google	Kendall's Tau B	—	0.351 ***
	p-value	—	< .001
Prediction (SLP) - Google	Kendall's Tau B		—
	p-value		—

		True Value (DCN) - Google	Prediction (DCN) - Google
True Value (DCN) - Google	Kendall's Tau B	—	0.764 ***
	p-value	—	< .001
Prediction (DCN) - Google	Kendall's Tau B		—
	p-value		—

		True Value (RFT) - Google	Prediction (RFT) - Google
True Value (RFT) - Google	Kendall's Tau B	—	0.762 ***
	p-value	—	< .001
Prediction (RFT) - Google	Kendall's Tau B		—
	p-value		—

		True Value (SLP) - Twitter	Prediction (SLP) - Twitter
True Value (SLP) - Twitter	Kendall's Tau B	—	0.377 ***
	p-value	—	< .001
Prediction (SLP) - Twitter	Kendall's Tau B		—
	p-value		—

		True Value (DCN) - Twitter	Prediction (DCN) - Twitter
True Value (DCN) - Twitter	Kendall's Tau B	—	0.766 ***
	p-value	—	< .001
Prediction (DCN) - Twitter	Kendall's Tau B		—
	p-value		—

		True Value (RFT) - Twitter	Prediction (RFT) - Twitter
True Value (RFT) - Twitter	Kendall's Tau B	—	0.727 ***
	p-value	—	< .001
Prediction (RFT) - Twitter	Kendall's Tau B		—
	p-value		—

Note. * p < .05, ** p < .01, *** p < .001

Table 4.01. Table of Kendall's tau-b coefficient for true and predicted turbulence values

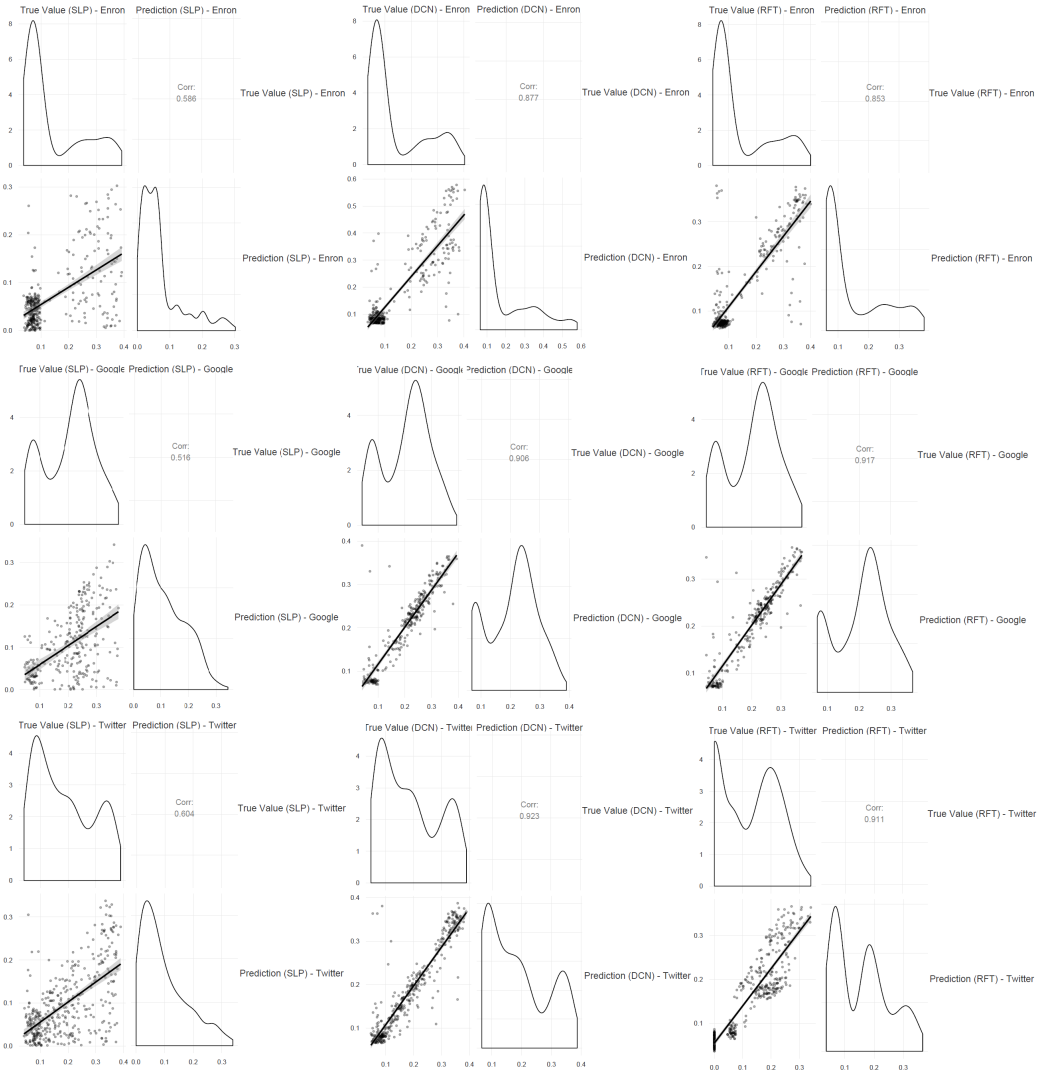


Figure 4.32. Graphs of Kendall's correlation, densities and statistics for true and predicted turbulence values

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.076	-0.077	-0.078
$\mathfrak{I}_i$	-0.805	0.764	0.002
$\mathfrak{Z}_i$	0.844	0.837	0.703
$\mathfrak{Q}_i$	0.901	0.872	0.871
$\mathfrak{L}_i$	0.877	0.913	0.953
$\mathfrak{J}_i$	0.788	0.827	0.891
$\mathfrak{S}_i$	-0.303	-0.297	-0.295
$\mathfrak{,Z}_i$	-0.271	-0.302	-0.312

Table 4.02. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Twitter validation dataset averaged over k cross validations (SLP)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.079	-0.074	-0.077
$\mathfrak{I}_i$	-0.783	0.776	0.001
$\mathfrak{Z}_i$	0.874	0.843	0.767
$\mathfrak{ב}_i$	0.892	0.882	0.846
$\mathfrak{L}_i$	0.864	0.891	0.921
$\mathfrak{ג}_i$	0.779	0.833	0.888
$\mathfrak{S}_i$	-0.293	-0.289	-0.278
$\mathfrak{ך}_i$	-0.275	-0.276	-0.284

Table 4.03. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Twitter validation dataset averaged over k cross validations (DCN)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.074	-0.070	-0.076
$\mathfrak{I}_i$	-0.738	0.825	0.007
$\mathfrak{Z}_i$	0.842	0.847	0.787
$\mathfrak{Z}_i$	0.887	0.834	0.837
$\mathfrak{L}_i$	0.846	0.884	0.901
$\mathfrak{Z}_i$	0.784	0.846	0.871
$\mathfrak{O}_i$	-0.285	-0.292	-0.269
$\mathfrak{Z}_i$	-0.273	-0.287	-0.278

Table 4.04. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Twitter validation dataset averaged over k cross validations (RFT)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	-0.012	-0.031	-0.048
$\mathfrak{I}_i$	-0.262	0.462	0.360
$\mathfrak{Z}_i$	0.344	0.338	0.303
$\mathfrak{Z}_i$	0.401	0.472	0.371
$\mathfrak{L}_i$	0.357	0.311	0.353
$\mathfrak{Z}_i$	0.378	0.327	0.391
$\mathfrak{S}_i$	-0.576	-0.602	-0.595
$,\mathfrak{Z}_i$	-0.514	-0.588	-0.542

Table 4.05. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Google validation dataset averaged over k cross validations (SLP)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.018	0.071	0.042
$\mathfrak{I}_i$	0.270	0.301	0.289
$\mathfrak{Z}_i$	0.316	0.348	0.333
$\mathfrak{Z}_i$	0.397	0.431	0.351
$\mathfrak{L}_i$	0.337	0.329	0.383
$\mathfrak{Z}_i$	0.362	0.367	0.321
$\mathfrak{S}_i$	-0.542	-0.611	-0.587
$,\mathfrak{Z}_i$	-0.533	-0.547	-0.556

Table 4.06. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Google validation dataset averaged over k cross validations (DCN)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.005	0.005	0.005
$\mathfrak{I}_i$	0.286	0.314	0.298
$\mathfrak{Z}_i$	0.414	0.337	0.345
$\mathfrak{Z}_i$	0.399	0.423	0.399
$\mathfrak{L}_i$	0.383	0.384	0.384
$\mathfrak{Z}_i$	0.392	0.379	0.377
$\mathfrak{O}_i$	-0.558	-0.607	-0.597
$\mathfrak{Z}_i$	-0.514	-0.551	-0.548

Table 4.07. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Google validation dataset averaged over k cross validations (RFT)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.581	0.435	0.437
$\mathfrak{I}_i$	-0.601	0.781	0.080
$\mathfrak{Z}_i$	0.781	0.978	0.883
$\mathfrak{Z}_i$	0.842	0.901	0.857
$\mathfrak{L}_i$	0.744	0.812	0.846
$\mathfrak{Z}_i$	0.891	0.789	0.861
$\mathfrak{O}_i$	-0.481	-0.387	-0.375
$\mathfrak{Z}_i$	-0.361	-0.359	-0.361

Table 4.08. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Enron's email validation dataset averaged over k cross validations (SLP)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.446	0.443	0.444
$\mathfrak{I}_i$	-0.708	0.762	0.070
$\mathfrak{Z}_i$	0.817	0.923	0.848
$\mathfrak{Z}_i$	0.831	0.947	0.837
$\mathfrak{L}_i$	0.747	0.851	0.875
$\mathfrak{Z}_i$	0.875	0.774	0.884
$\mathfrak{O}_i$	-0.451	-0.348	-0.344
$\mathfrak{Z}_i$	-0.377	-0.374	-0.375

Table 4.09. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Enron's email validation dataset averaged over k cross validations (DCN)

Spearman (rho) coefficient			
	$P(\gamma_{rl})$	$P(\vartheta_{rl})$	$P(\varphi_{rl})$
$\mathfrak{C}_i$	0.435	0.433	0.434
$\mathfrak{I}_i$	-0.750	0.728	0.075
$\mathfrak{Z}_i$	0.834	0.930	0.834
$\mathfrak{Q}_i$	0.849	0.974	0.878
$\mathfrak{L}_i$	0.784	0.818	0.811
$\mathfrak{z}_i$	0.817	0.725	0.891
$\mathfrak{S}_i$	-0.402	-0.322	-0.319
$,\mathfrak{I}_i$	-0.359	-0.357	-0.358

Table 4.10. Table of Spearman's (rho) coefficient between independent input and dependent output variables for the Enron's email validation dataset averaged over k cross validations (RFT)

K	δ_{MAPE} - (SLP)	δ_{MAPE} - (DCN)	δ_{MAPE} - (RFT)
20	0.461	0.189	0.127
30	0.424	0.175	0.131
50	0.420	0.173	0.112
80	0.418	0.169	0.110
100	0.421	0.166	0.107

Table 4.11. Table of K-fold cross validated MAPE for all three learning models

CHAPTER 5

EVENT PREDICTION USING FRACTAL NEURAL NETWORKS

5.1 Introduction

Event prediction is a complex topic which leverages techniques from multiple disciplines across wide ranging applications [2]. Some of them include recommender systems, marketing and advertising, governance and rule, news and propaganda, etc. [3]. The social pre-cursors of a large majority of real life events are often staged through popular online social media like facebook, twitter, google, etc. These pre-cursors are often identified as activity through online social mediums as information transactions [4]. Although it may be intuitive to think of a similarity based approach on how an actor influences other members within a community through matching attributes, such an aggregation of affective sentiments are often times a lot less direct [5].

In the previous chapter, we tackled the problem of describing relational flux and turbulence of three well-established major social networks (google, twitter and enron emails) using principles of the Relational Turbulence Model (RTM) [12]. In this chapter, we study ground truths proposed by the Relational Turbulence Theory [9] and adapt it to uncover evolutionary social transaction behaviors for event prediction. We also further develop our novel Fractal Neural Network (FNN) learning architecture to scale towards predicting different events through a series of temporal relational transactions in a vast social environment constantly evolving with sentimental and affective disruptions with topic drifts [13], [14]. This improvement in performance and accuracy is demonstrated in the results and discussion section of this chapter. Some examples of emerging event prediction applications include: Pre-emptive disease and medical condition prevention, patient-drug matching pair diagnosis and administration, cyber security, data privacy and utility,

etc.

In this study, we improve on our model, the Relational Flux Turbulence (RFT) using principles of the Relational Turbulence Theory (RTT) [9] to establish a framework of theoretical processes linking evolving relational features learned over past event occurrences (causals) to relational reciprocity bias, sentimental and affective communication patterns, state altering events and role-recognition behaviors that identifies relational uncertainty and interdependence [21] as parameters in correlation to generalized event occurrences on three major streaming social platforms: Twitter, Google Feed and Live Journal.

Past event occurrences from a stream of interesting social transactions were first filtered by Geo-Locality [246]. A progressive threshold wavelet-transform variant with adaptive scaling and shift sampling was developed to detect eventful occurrences from an archive of mixed social information transactions [247]. Latent Dirichlet Allocation (LDA) Mixture Membership Model (MMM) based segmentation and clustering was performed on the established tree of key words to group them according to topics in a general query stream of data [248], [124], [123]. The improved RFT model then incorporates the use of an integrated adaptive LSTM convolutional block to feedback information learned from these features retained in past event occurrences back into the fractal neural network base architecture to improve performance of its meta-learning evolutionary states.

Our model accepts as inputs, concurrent key relational feature states f_i between actors E_ϵ from past and present social transactions to predict the likelihood of an event occurance E_φ in an evolving state of relational turbulence τ_{ij} from an identified social flux F_ϵ within a continuous stream of social transactions [27], [28]. Relational turbulence may correspond to various disruptions in social communication of different environments and contexts [9]. For example, in the discussion of a world event like trade wars, passive sentiments passed through public posts and comments are indicative of hostility and potential conflict which may lead to a breakdown of linked integrity between actors in many aspects like trust, influence, status, etc. [63]. In addition, as a major contribution of this chapter, we also show that

the RFT model improves efficacy of existing approaches towards event prediction through studies and comparisons of experimental results conducted on real life social networks. Then, we evaluate our methods on Twitter, Google and LiveJournal datasets and demonstrate that they outperform hybrid Probabilistic-Markovian (PR) based future event prediction models [1].

Event detection, remains one of the most challenging and interesting areas of development in social networks today. Although numerous approaches have been developed to address certain areas of effectively detecting events, their methods have been limited in application to specific events in question [4]. Furthermore, approaches to date have been focused on the use of batch learning methods which can only be used at static instances in time [17]. Such approaches have been known to be notoriously unscalable to continuous data streams and changing environment contexts [14]. Thus, several critical key questions in this field of study remain unanswered. In an unstructured and generalized social network of actors connected to each other by an unconstrained and evolving construct of Markovian relationships which are changing through time [61], how can we firstly, efficiently and effectively represent the generalizations of the evolutionary behavior within these social transactions? Secondly, how accurate are predicted future events in correlation to relational features of their occurrences? Finally, how can we quantify the dynamic errors arising from social disruptions and topic drifts (outliers) in our predictions? We answer these questions with the use of fractal neural networks, which encode the RTT framework ground truths into the lowest principle decompositions of our model as a means to self-evolve from a meta-learning perspective - in response to random "anytime-sequenced" data batches of fluctuating information sophistication [19].

In this chapter, the RFT (Relational Flux Turbulence) model for event prediction will be presented, which is developed from the principles of self evolving fractals and artificial neural networks in a real-time machine learning model [26], [18]. Its objective function describes the turbulence profiles of social graph constructs and their resulting communication behavioral patterns across apriori relational state altering events - to predict likelihood occurances of tracked topics as events of interest. The scientific contributions of our work involve the following:

1. The method adaptively learns from a Fractal Neural Network (FNN) which builds on key relational fractal structures discovered in a given Online Social Network (OSN) from tracked topics. As an improvement to conventional event tracking and detection approaches, our method is able to handle topic drifts seamlessly in real-time within a given Mixed Membership Model (MMM) corpus of social transactions (e.g. tweets, google feeds, blogs, etc.);

2. An innovative RFT model was developed to capture key relational features which were used to train relational fractals within a given topic-event context. The relational turbulence profiles of communicative state patterns observed between known actors in an event-related community of past event occurances were then used to build an FNN framework architecture to accurately and efficiently predict future occurances of similar events;

3. Experiment results show that RFT is able to offer a good modeling of relational ground truths, while FNN is able to efficiently and accurately predict likelihoods of event occurances.

The remaining part of the chapter is organized as follows: Section 5.2 presents a brief overview of related works and introduces key concepts drawn from social theories and relational structures. Section 5.3 introduces the theories and methods of our proposed model. Section 5.4 provides a thorough analysis of experimental design and implementation. Section 5.5 presents the results and discussion of this chapter that leads to a conclusion and potential future directions.

5.2 Related Literature

Relational Turbulence was first studied as an observation in [20] to characterize the behaviors of communication between social actors at stages in an environment of progressive relational developments. This study recorded the polarization of sentiments and reaction reciprocities between the actors in an environment of constant social shocks. The observations were then first

conceptualized from conducted experiments as a black box model known as the Relational Turbulence Model (RTM) [15]. The RTM anchored on relational uncertainty and interference as key output features which are directly correlated to relational turbulence that increases during relational state transitions which have shaped apriori information bias. Some empirical model tests include the cross-sectional self-report methods [9], the longitudinal self-report methods [9], laboratory observations, recordings of dyadic interactions [21] and theme analyses of discourse [21].

While the RTM is adequate in offering a cause and effect framework for modeling relational turbulence, it is lacking in three substantial areas. The RTM does not offer distinctive processes through which key relational reciprocal features arising from actor uncertainty and interference affect the evolution of relational communication behaviors [9]. Secondly, RTT establishes correlations between a subset of causal relational features to observed sentimental and affective social transaction behaviors which is missing in the RTM framework [9]. Thirdly, RTT establishes a Markovian construct where specific signature evolution patterns of graphical social transactions are correlated to their corresponding detected event occurrences whereas the RTM merely models the time specific relational turbulence profile within an identified relational flux [21].

In [9], the authors study the extensions between relationship parameters, episodic experiences and outcomes of cumulative effects. The authors argue that reciprocal effects caused by biased cognitive appraisals of sentimental stimulus during social exchanges causes variations in communication patterns between actors. These variations are often amplified during state altering event transitions. In their paper, relational uncertainty is broadly defined as the sentimental polarity exhibited in between social exchanges while interdependence (interference) is characterized as the dynamic threshold of allowable influence perceived at a given state of an evolving relational flux. In this respect, their proposition is that influence is positively correlated to interference. This means that the more influence actor A is able to exert on B (high social interdependence thresholds), the more likely A is able to interfere with Bs social communication patterns and vice versa. This also means that key interdependent relationships exhibiting high levels of influ-

ence within a social construct are now more capable of creating turbulent communication patterns and behaviors precedent to an event occurrence.

Event prediction itself has been vaguely explored from a social front. Most recent work done on predicting events rely on probabilistic inference mechanisms of atypical event sequenced information from given textual contexts.

In [2], the authors tackle the problem of prediction of events as a time probabilistic uncertainty. Their developed genetic algorithm timeweaver uses the multiple-instance learning approach to learn representative events (past occurrences) in order to predict future (target) events. As a pattern recognition problem, their main approach focuses on observing patterns from traces of existing events to form a prediction of target events. The authors tackle prediction ambiguity by defining recall based on target events, rather than on the predictions themselves. They used individual patterns identified by their timeweaver algorithm to predict event subsets with high accuracy and collectively, these patterns cover most of the target events in question. Although their developments are good in may respects, they contain some flaws which needed to be addressed. As a multiple-instance learning approach, their method is unable to handle the presence of false positives and negatives from a bag of samples. This can easily lead to the problem of undersampling and ill-conditioning due to noisy samples at the learning input. Furthermore, their algorithm models the prediction of an event as a probabilistic step function. This means that a positive prediction of a future event is probabilistically represented as 100%. Therefore, they lack the representational capability to represent the predictions of future events as a time varying probabilistic distribution profile which is a more realistic representation of prediction patterns towards future events.

In [249], the authors addresses the problem of predicting future events through an archived database of sequenced events. Their formulation of sequential event prediction is derived from supervised ranking applications like recommender systems, equipment maintenance, medical informatics, etc. Such applications focuses on the predictive power of past event sets instead of their specific order which in turn leads to differential sequenced event problems and algorithms. Their algorithm treats each step of the sequential event

prediction problem as a supervised ranking objective function. This means that with a given subset of observable events, their approach ranks all other events in order of maximum likelihood occurrences as subsequent events from the bag. Their general framework hinges on the concept of minimizing risks from empirical event observations. The authors present two scoring models to achieve this they are the one-stage model and the ML-constrained model. The difference between the scoring models is that the one-stage model relies on unconstrained real-valued variables to determine the probability of influence that events in set A has on an identified event sequence B. Whereas the ML-constrained model reduces this cardinality by conditionally constraining the probability of a future event B occurring in sequence, given that events in set A has occurred in their corresponding sequential pattern/s. In their experiments, they compare the performance of the ERM-based algorithms to the max-confidence association rule and the item-based collaborative filtering methods. Their experiments were conducted over three real life event prediction applications. They are the Email recipient recommendation, the patient condition prediction and the online grocery store recommender system. Although their ERM algorithm outperformed the max-confidence and cosine similarity baseline predictions, they are not scalable to large scale topic detection and event prediction problems like OSNs. As a loss minimization function approach, an objective function is to explore all posterior spaces of event orderings in a given manifold. This can easily result in the problem of overfitting and poor conditioning of search functions, especially if the search spaces are non-convex.

In [1], the authors tackle the problem of event prediction using a hybrid probabilistic and time-series model approach to utilize off the shelf Information Retrieval (IR) systems into event predictors. The authors used a topic based approach to define an event as an indirect observable incident influencing a common interest within a socially public context. Their approach consists of five key steps: 1. Information Retrieval, 2. Time-Series Classification, 3. Event-Peak Detection, 4. Probabilistic Model Training and 5. Prediction. They conducted experiments on the New York Times corpus and show that hybrid models outperform baseline prediction methods like Support Vector Machines (SVMs), BNets, etc. in reducing prediction error. They achieved this by translating the retrieved information into in-

direct bursty time sequenced event signals. The periodicity of these time signals were then extracted using an autocorrelation function and an estimated probabilistic model for predicting future (event) signal peaks. Although their models were well defined and adequate at classifying time series data and peak predictions, they lack representational power to fully describe the time-evolution of the signal preceding an event. Furthermore, they are constrained dimensionally to time sequencing of events, which in real life, may not always be a natural occurrence. This means that certain events remain independent of each other and do not necessarily have to occur in sequence after each other. Such a model will not be able to accommodate the degree of randomness in their corresponding event signal peaks.

In [25], the authors tackle the problem of firstly acquiring knowledge sequences from text and secondly, developing a predictive model for use in narrative generation systems. Their model first adopts the multiple choice narrative cloze task to extract the likelihoods of ranks between current topic contexts and a vocabulary set of subsequent events in the word chain. In order to keep things simple, they use latent semantic indexing (LSI) to derive a vector representation of events in terms of the contexts from which they were last seen to act as a baseline for other vector-space models. They retrieve embeddings of verbs using a Word-2-Vec method in order to provide a suitable measure to judge the correlations between two events. Next, a compositional neural network model accepts as inputs, the predicates and arguments of the Word-2-Vec extractions of 2 event corpus and learns a non-linear sequential likelihood representation of the two events occurring from within the same chain. The authors experimented their model on the New York Times portion of the Gigaword Corpus and proved that their approach outperformed the positive pointwise mutual information (PPMI) measure. However, a flaw in their approach is that events predicted in this manner lack the social influence in predicting how and in what sequence most real life events unravel. Thus, although their model achieves good MCNC prediction accuracies from a fixed corpus perspective, it is still unable to represent event predictions as real life likelihoods of sentimental and affective social turbulence and unrest.

5.3 Theories and Methods

We begin our approach with the definition of events as bursty time-scaled periods of highly intense, dense and volatile social transaction/s within a given Online Social Platform [250]. These multi-dimensional peaks (e.g. frequency, sentiment, polarity, reciprocity, etc.) arising from relational turbulence in an online social scene due to the occurrence of an event carries a unique signature pattern defining the stretch and length to the profile of the observed burstiness in information exchange within a given social network [247]. From a practical viewpoint, a wavelet signal structure of an event can be used to match a real-time information exchanges in an active stream efficiently [247]. However, when used to predict events, it is unable to fully capture and represent the affective sentiments across known event priors adequately enough in order to accurately predict likelihoods of future occurrences [45]. Furthermore, a key assumption we make in this chapter is that the order of events are randomly distributed over the sentiments expressed in any given OSN/s. This key assumption derives from the fact that most real-life events are weakly dependent on each other from a sequential occurrence standpoint [28], [24]. Instead, they are highly correlated through key reciprocated relational sentiments to their common topic supersets of interest [4].

As our first step implementation, topic detection is done from a contextual corpus of words. The topic model developed in this chapter provides an exploratory analysis into large text corpora by learning the thematic structure of key vocabulary word embedding [251]. For accuracy, we have adopted the non-parametric mixture model as our statistical inference mechanism to deduce likelihoods of the underlying topic-word distributions and reject anomalous syntactic word-topic combinations [92]. Although it is noteworthy to point out that various other word distribution models like Latent Dirichlet Allocation (LDA), correlated topic models, Pachinko allocation, etc. may be assumed and used as drop-in replacements [124], [125], [123]. After topics have been detected from a general query data stream, continuous wavelet transformation is then used to uncover unique localized predicate signals for key anchor words in a time-scaled domain [4]. Once this signal has been decomposed into its linear basis functions, peak detection is then performed on these topic mention frequencies [247]. Then, an LDA inference mech-

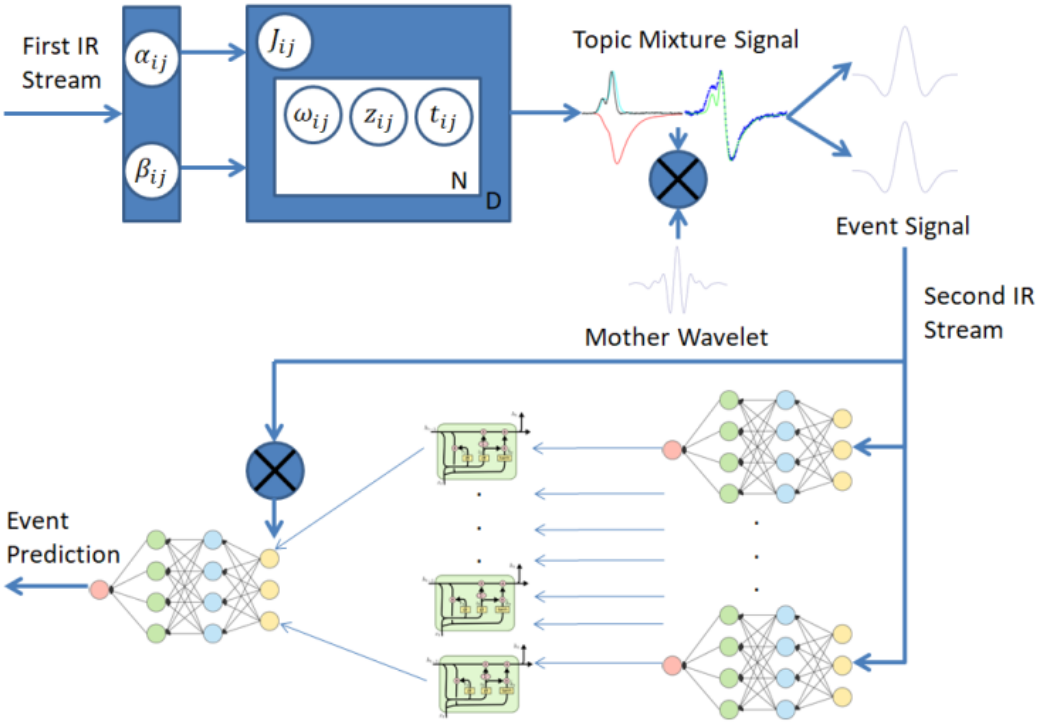


Figure 5.01. Event Prediction System Architecture. This diagram shows the key design features of the Event Prediction system which contains the RFT as a core module for recognizing relational turbulence.

anism based on the Gibbs sampling approach is then used to detect and identify events from the detected topics in the first IR text stream [123]. Once the events have been identified and their unique contextual key word signatures learnt, these events are then re-queried in a separate data stream. The continuous information flow received from this query allows us to focus our application of the RTT framework onto specific detected events of interest. We define the Relational Intensity $P(\gamma_{rl})$, Relational Interference $P(\vartheta_{rl})$ and Relational Uncertainty $P(\varphi_{rl})$ as causal relational features which represent Relational Turbulence $P(\tau_{rl})$ 1. that are correlated to observed sentimental and affective reciprocities of a given link in an OSN. The likelihood occurrence of the queried event is then positively correlated to the aggregated contribution of all detected relational turbulence in a continuous stream of social transactions within the constrained geo-locality of interest. A detailed structure of our approach is given in Figure 5.01.

The Relational Turbulence $P(\tau_{rl})$ of a given link in an OSN is determined key features of an established relationship in any instance. They

are the confidence ρ_{ij}, salience ξ_{ij} and sentiment λ_{ij} scores in a dyadic link. Their reciprocities are therefore - ρ_{ji}, ξ_{ji} and λ_{ji} respectively. It is intuitive to think that in theory, reciprocities in an actual data stream should match so as to reflect the true probabilistic measure of the potential occurrence of an event. However in reality, reciprocities are oftentimes expected to violate actor expectancies. Shared expectancies are defined when $E(\rho_{ij}, \xi_{ij}, \lambda_{ij}) \equiv E(\rho ji, \xi ji, \lambda_{ji})$. Non-shared expectancies are therefore non-uniformly threshold expectations of relational reciprocals. This means that $E(\rho_{ij}, \xi_{ij}, \lambda_{ij}) \neq E(\rho_{ji}, \xi_{ji}, \lambda_{ji})$. Burst expectancy violations (EV) are measured as the mean deviation over a sliding window of information exchange through a subset of social transactions. They are contributing factors to temporal representations of relational turbulence - γ_{rl}, ϑ_{rl} and φ_{rl}.

Consequently, polarities of violations are defined as the threshold mismatch between expected and actual mismatch of reciprocates [9]. In our chapter, this measure is represented as a vector where violations have both a signed magnitude and direction (ingress and / or egress). Cosine similarity is used to determine both positive and negative EVs within a given stream of reciprocates. A time-scaled event of $P(\tau_{rl})$ is often characterized by sharp and frequent EV peaks where gradient change of their weighted feature scores are high. This is given mathematically as:

$$\frac{\partial E_{rl}}{\partial \tau_{rl}} = \sum_{i,j=1}^{n} \prod_{\eta=\rho,\xi,\lambda} \frac{\partial E(\eta_{ji})}{\partial \eta_{ji}} \times \frac{\partial \eta_{ij}}{\partial \tau_{ij}} \tag{5.1}$$

Where τ_{ij} is also known as the relational turbulence between node i and its surrounding neighbors j. Thus, $\frac{\partial E_{rl}}{\partial \tau_{rl}}$ is also known as the communication valance.

Relational state transitions are defined as state-based critical violation frequency thresholds, beyond which relational turbulence and negative communication profiles become irrevocable [21]. This critical threshold is dyad specific and learned through our model as a conflict escalation minimization function. Conflict escalation is defined as the gradual increase in negative relational flux $-\frac{\nabla F_\epsilon}{\nabla t}$ over time within a classified context area L_{F_ϵ} (event query) of interest. The critical threshold parameter is then driven mathematically as:

$$T_\epsilon = inf_{t\to\infty} \begin{cases} \frac{1}{2m}(-\frac{\nabla F_\epsilon}{\nabla t} \times \log_2(\frac{\nabla F_\epsilon}{\nabla L_{F_\epsilon}})) \\ \log_2(|1 - \frac{\nabla F_\epsilon}{\nabla L_{F_\epsilon}}|) \end{cases}$$

Where T_ϵ is the threshold of interest and m is the total number of training data over the time window t.

We define relational intensity as the continuous integration of sentimental transactions per context (event topic) area, the relational uncertainty as the likelihood from opposing sentiment mentions and relational interference as the probabilistic deviations in expectancies from predicted uncertainties and flux intensities. Mathematically, these are given as:

For Relational Intensity:

$$\gamma_{rl} = \sum_{i,j=1}^{n} \frac{\beta_{ij}| - \frac{\nabla F_{\epsilon j}}{\nabla t}|}{L_{F_\epsilon}} + \chi_{rl} + \dot{\theta}_{rl} \tag{5.2}$$

Where β_{ij} is defined as the temporal derivative of the latent topic (context) oscillation phase ϵ, χ_{rl} is the reciprocal bias and $\dot{\vartheta}_{rl}$ is the gradient of social influence from one actor to another across a relational link.

For Relational Uncertainty:

$$\varphi_{rl} = \frac{\sum_{i,j=1}^{n} S_i S_j}{\sqrt{\sum_{i=1}^{n} S_i}\sqrt{\sum_{j=1}^{n} S_j}} \tag{5.3}$$

Where S_i and S_j are sentiments transacted from nodes i to j and from nodes j to i respectively.

For Relational Interference:

$$\begin{aligned} \vartheta_{rl} &= E(F(\gamma_{rl}, \varphi_{rl} : \mu_{\gamma\varphi}, \omega^2_{\gamma\varphi})) \\ &= \tfrac{1}{2} + \tfrac{1}{\sqrt{2\pi}\omega} \sum_{\gamma_{rl},\varphi_{rl}=0}^{n} \tfrac{1}{2} erf(\tfrac{\gamma_{rl},\varphi_{rl}-\mu}{\sqrt{2}\omega}) \exp^{-\frac{(\gamma_{rl},\varphi_{rl}-\mu)^2}{2\omega^2}} \end{aligned} \tag{5.4}$$

Where,

$$F(\gamma_{rl}, \varphi_{rl} : \mu_{\gamma\varphi}, \omega^2_{\gamma\varphi}) = \frac{1}{\sqrt{2\pi}\omega} \sum_{t=-\infty}^{\gamma_{rl},\varphi_{rl}} \exp^{-\frac{(t-\mu)^2}{2\omega^2}} \, dt \tag{5.5}$$

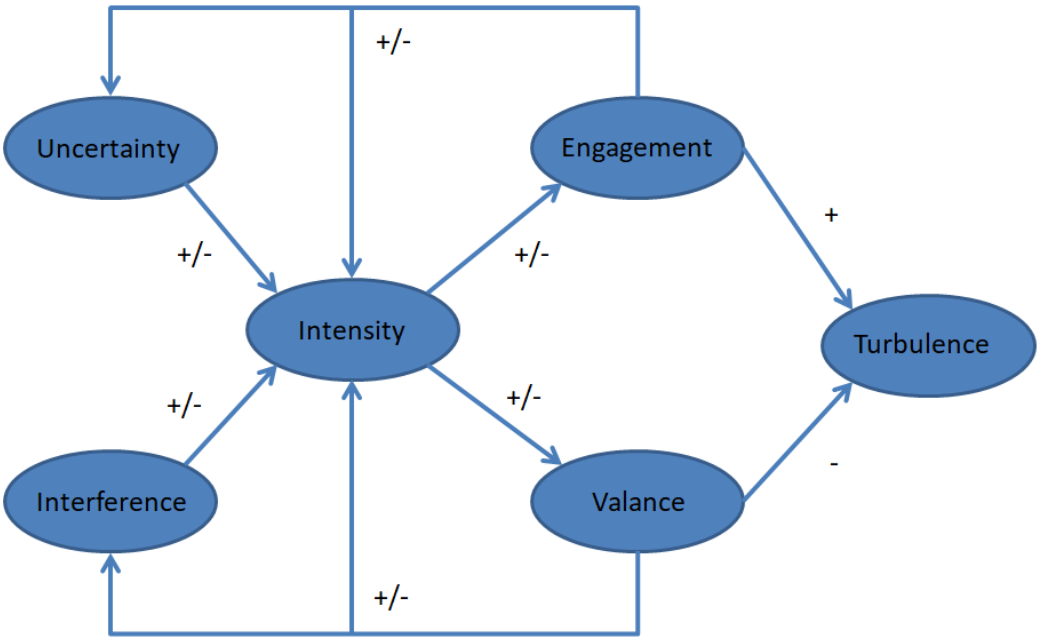


Figure 5.02. The simple RTT framework. This diagram shows the over-simplified logical connections between social states during instances of information transaction/s between actors in an OSN.

Here, $F(\gamma_{rl}, \varphi_{rl} : \mu_{\gamma\varphi}, \omega^2_{\gamma\varphi})$ is the Cumulative Distribution Function (CDF), and $erf(x)$ is the error function of the predicted outcomes γ_{rl} and φ_{rl}. This can be represented as:

$$\chi_{rl} = \sum_{\gamma_{rl},\varphi_{rl}=0}^{n} \frac{1}{2} erf(\frac{\gamma_{rl}, \varphi_{rl} - \mu}{\sqrt{2}\omega}) \tag{5.6}$$

A dyadic relational turbulence is characterized by the mixed contribution model of all three key feature attributes of a relationship. We define the dyadic relational turbulence model to be constrained by the RTT framework where relationship parameters of dyadic actor uncertainty and interference contribute to episodic a-priors of relational intensity, communication polarity, communication engagement and reciprocal bias. This representation is given in Figure 5.02.

Communication engagement is defined mathematically as:

$$\varepsilon_{rl} \in Hom(\amalg_{\tau \in T} \chi_{ij}, \gamma_{ji}) \tag{5.7}$$

Such that the communication engagement ε_{rl}, represents the unique isomorphism $\varepsilon_f : \chi \to \gamma$.

Finally, we adopt the RFT model to learn the structure of the fractal neural network represented by the RTT framework to predict the likelihoods of future event occurrences in question.

5.3.1 Topic Detection

Topic modeling falls into two broad categories of approach: The generative model and the stochastic model [124]. Generative models use word distribution kernels over topic mixtures generated by the document. Such a model defines word co-occurrences to be mutually inclusive over the generative topics of a given corpora. This means that mathematically:

$$P(W_i \cup V_j) = \sum_{i,j \in D} P(W_i) + P(V_j) - P(W_i \cap V_j) \tag{5.8}$$

Where co-occurrences of words W_i and V_j for all i, j belongs to the document corpora D is distributively polysemic. The main assumption which this model class draws on is that likelihood occurrences of words (word frequency) in a target corpora are essentially distributed according to the probability densities associated with the convolution of their defining weighted topic distribution mixtures [251]. This assumption is also known as the bag-of-words assumption and is most popular in Latent Semantic Analysis (LSA) among many other statistical masks. The main drawback of this model however, is that by reducing document-word features into a convoluted distribution, Markovian word appearance information is discarded. This means that syntactic word choice information is not well handled using this approach. Several other methods have since been developed to tackle this area of research. They include: Word-2-Vec, Cosine Similarity, Topic Segmentation, Structured Distribution, Compound Topic Model, etc.

The stochastic model however, statistically inferences topics to words in a given corpora from observations of words over a set of key documents to determine posterior likelihood estimations of topics over documents, words

over topics and the most prominent topic model used to have generated each word in question. Essentially, this can be mathematically expressed as the joint probability:

$$P(k|W_{d,u,n}) = \frac{P(W_{d,u,n} \vee k)P(k)}{\sum_{s \in k} P(W_{d,u,n} \vee k)P(k_s)} \tag{5.9}$$

Where k is the topic assignment of the word $W_{d,u,n}$ in document d over the word passage u with a co-occurance word count n and s is the topic segment in question. This approach makes no real assumptions about how distributions over topics and / or their word frequency correlations are masked. But instead estimates these posterior distributions from apriori observations [92]. Some developments in this area include: Markov Chain Monte Carlo (MCMC), Gibbs Sampling, Metropolis Hastings, etc. An added advantage of stochastic approaches is that it is capable of establishing posteriors which are as close as possible to real-time estimates of topic hierarchies.

In our chapter, we have used the non-parametric mixture model to detect topic models over a continuous time stream of social exchanges on Twitter, Google Feed and LiveJournal. This enables our model to effectively tackle the problem of topic drifts and establish soft event footprint evolution over time.

Given a continuous stream of contextual information exchanges forming the corpora, we assume that each time-batched social transaction $d_j \in D$, contains a unique set of tokens $\zeta_i \in Z$ over a Hierarchical Dirichlet Process (HDP). Each token atom is defined to be constructed from an ordered pair of word-time primitives. Mathematically, this is expressed as:

$$\zeta_{ji} := (W_{ji}, t_{ji}) \tag{5.10}$$

We assume that the temporal variation of each word-topic pair follows a multi-modal distribution given mathematically as:

$$P(k_{n,t}|\zeta_{n,t}) = (\sum_{s=1}^{S} P(s \vee \zeta_n)P(k_{nt} \vee \zeta_{nt}, s)) \tag{5.11}$$

Where $P(k_{n,t}|\zeta_{n,t})$ is the conditional probability that topic k (an unknown prior) is chosen for the token ζ. Essentially, this means that the probability that topic k has been chosen from the token word frequency n, over an observation time window t, is conditionally dependent on the characteristics of the token $\zeta_{n,t}$ in that same observation [252]. This conditional probabilistic dependence is equal to the sum total over all possible topic segments over which time independent token characteristics ζ_n is a member of, given a posterior topic selection for the token atom.

The likelihood estimator for the topic assignment is then given mathematically as:

$$\Lambda = \prod_{n=1}^{N}(\oint_{\eta}\sum_{s=1}^{S}(P(s|\zeta_n)P(k_{nt}|\zeta_{nt},s,\eta))f(\eta)d\eta) \tag{5.12}$$

Where the error function η is given mathematicall as:

$$\eta = argmin_{\delta}\int_{k\in T} R(k,\delta)P(k)dk \tag{5.13}$$

Here, δ is the decision rule and $k \in T$ indicates that topic k belongs to the corpus of topic superstructure T. $R(k,\delta)$ is the risk function associated with the decision rule δ driven mathematically as:

$$R(k,\delta) = \int_{\zeta} L(k,\delta(\zeta))dP_k(\zeta) \tag{5.14}$$

Where $L(k,\delta(\zeta))$ describes the loss function consequent from having chosen topic k based on decision rule δ characterized by the social transaction token atom ζ.

Our topic inference mechanism is based on a Markov Chain Monte Carlo (MCMC) estimation process built around HDP mixtures [92] to efficiently label topics according to a Dirichlet distribution process over a shared relational hierarchy of parent-child topic segments. MCMC estimation is used as a more generalized form of stochastic approximation which yields relatively good results from good initializations over flat search spaces. However, search performance does degrade significantly if regions of localized minimas exists within the search spaces. For that reason, monte carlo variants like the HMC,

Metropolis Hastings, reduced gibbs sampling etc. can be used as drop-in replacements for MCMC.

We establish that each token atom ζ, is associated to an atom specific decision $\delta(\zeta)$ - which maximizes likelihoods of a topic assignment to a token and minimizes errors of the relationship made with the label. In turn, each $\delta(\zeta)$ is correlated to the word observed from the document-token indexed pair. Additionally, each $\delta(\zeta)$ is also correlated to the time window of the document-token indexed observation pair. Thus mathematically, this is given as:

$$\zeta_{ji} :\rightarrow \begin{cases} \cong G(W_{ji}) \\ \cong G(t_{ji}) \end{cases}$$

Where G approximates to a dirichlet process with scaling factor α_0 and a base probabilistic measure (also dirichlet distributed) G_0.

Thus, we sample from 2 directly correlated dirichlet distributions over a continuous stream of data flow.

We represent the sampling τ_{ji} over the time dirichlet distribution process $G(t_{ji})$ by writing it mathematically as:

$$P(\tau_{ji} = t_{ji} | t^{-ji}, k) \propto \begin{cases} n_{W_{ji}}^{-jt} \text{In a previous time step} \\ \alpha_0 \text{Otherwise} \end{cases}$$

Likewise, for sampling ω_{ji} over the word dirichlet distribution process $G(W_{ji})$ can be represented mathematically as:

$$P(\omega_{-ij}^{W_{ji}} = k | W^{-ji}, \zeta) \propto \begin{cases} m_k^{-jt} \xi_k^{-\zeta_{jt}}(\zeta_{jt}) \text{For previous topic mentions} \\ \gamma_0 \xi_k^{-\zeta_{jt}}(\zeta_{jt}) \text{Otherwise} \end{cases}$$

Where m denotes the topic mention frequency and γ_0 is the scaling factor of the word dirichlet distribution.

5.3.2 Wavelet Transform and Event Detection

Wavelet transformation is used in our model to identify events after a topic mixture has been successfully sampled with from the previous step. Since each topic category already contains words with high likelihoods of association with it, we are only interested in separating specific events from general themes of discussion from our mixture. The core feature of wavelets and their transformations is to enable analysis of signal profiles at specific time scales from the full time window of the document-token indexed observation pair t_{ji}. This is done over all tokens ζ_i of the social transaction batch d_i of interest. The resolution of this transform is adaptively adjusted according to approximations of sharp discontinuities from forest (wide window) to trees (small window). These approximations allow us to separate specific events from general themes of discussion.

We start with defining a mother wavelet $\psi(t_{ji})$, and construct child wavelets by adaptively determining scaling and translation factors ν and ϕ respectively. This means that mathematically, a wavelet family can be represented as:

$$\psi_{\nu,\phi}(t) = \frac{1}{|\sqrt{\nu}|}\psi(\frac{t-\phi}{\nu}) \tag{5.15}$$

Where $\nu, \phi \in \Re$ and $\nu \neq 0$.

Wavelet transformations fall into two broad categories. They are the discrete and continuous transforms [4]. While the CWT enables smooth detection of slow and continuous varying features, DWT provides a more efficient mechanism of detecting discontinuous signals [247]. In our architecture, we have adopted the DWT to overcome the problem of solving for an infinite number of coefficients which is computationally intensive. Furthermore, the resulting transform of DWT is unit orthogonal to each other. This means that with a finite space parameter on one plane $\nu_q = 2^{-q}$, the other parameter can be expressed as the cross product, $\phi_r = 2^{-q} \times r$ to define an inner product space for the discrete wavelet as:

$$\psi_{q,r}(t) = 2^{-q/2}\psi(2^{-q}t - r) \tag{5.16}$$

Finite orthonomality between signal coefficients also guarantees that the original $f(t)$, may be reconstructed after decomposition. This is mathematically written as:

$$F(t) = \sum_{q,r} E_q(r)\psi_{q,r}(t) \tag{5.17}$$

Where $E_q(r)$ are residual wavelet error coefficients of progressive signal approximations at varying time scales of q. Since discrete child wavelets are replicated from a mother and are finite, it intuitively means that the original signal is simply the sum of all wavelets in the decomposition scale.

$$F(t) = \sum_{q} f_q(t) \tag{5.18}$$

Provided that the sample frequency of the constructed wavelet falls within Nyquist criterion, $2^{-q}\omega_s \leq |\omega| \leq 2^q\omega_s$.

The Shannon Wavelet Entropy of the original signal $F(t)$ is given as:

$$En(F(t)) = -\sum_{q} \rho_q log\rho_q \tag{5.19}$$

Where ρ defines the relative wavelet energy at different scales.

$$\rho_q = \frac{E_q}{E_{Total}} \tag{5.20}$$

The wavelet classification H-measure of the signal $F(t)$ is given as:

$$H(s) = \frac{En(F(t))}{En_{max}} \tag{5.21}$$

The token-topic signal in a given time scale can be reconstucted as:

$$S_k(t) = \frac{N_k(t)}{N(t)} \times \log \frac{\sum_{x=1}^{T} N(x)}{\sum_{x=1}^{T} N_k(x)} \tag{5.22}$$

Where $N_k(t)$ is the number of tweet messages which contain the token ζ referenced by the topic k and appears within the time window $(t-1 \leq t \leq t+1)$. $N(t)$ measures the total number of tweets within that same time window. The fraction $\frac{N_k(t)}{N(t)}$ denotes the salience of a topic k over the sampling time

scale of interest. While the second term on the right measures the inverse log-linear relationship between the number of times a topic has been referenced to from a token taken as the target observation. Essentially, the second term acts as a linear scaling factor to filter out false peaks. This means that for topics which have less mentions within a sample window of tweets are disregarded as general discussion themes (with low signal - S_kt peak scores) while topics with more mentions within that same sample time window are identified as potential occurring events (with high signal - S_kt peak scores) within the same time window in question.

Signal entropy is a measure of how power within the signal is distributed over a time frame of observation [247]. The intuition that follows this logic is that as we keep observing across successive time windows, we would like to know if the topic signal which we have constructed from the given time scale spreads out over time. Thus, the larger the spread of this signal, the more sparse the distribution of power over larger time frames. Conversely, the smaller the spread of this signal, the more dense this signal power distribution is on a focused time span. Therefore, it can be inferred that the larger the entropy of a signal over successive time frames of observation, the more likely that topic belongs to a general theme of discussion; while the smaller the entropy of a signal, the more likely that topic is classed as an event which had occurred within a specific time frame. Since events are characterized by short bursts of token-topic mentions over a time window of tweets, we can easily identify if a topic belongs to an event or general discussion theme by analyzing both signal peak scores S_kt and entropy changes of a wavelet [250], [81]. Since the H-measure is the normalized form of entropy measure which can be easily used across varying time windows [84], [141], we adopt it as a measure to identify changes in token-topic wavelet entropy across longer observation time frames [253]. This is given mathematically as:

$$\triangle H(S_k(t)) = \begin{cases} \frac{H_{t+\triangle} - H_{t-1}}{H_{t-1}} \text{For } H_{t+\triangle} > H_{t-1} \\ 0 \text{Otherwise} \end{cases}$$

An example plot of signal peak scores $S_k(t)$ across a sliding time window is given in Figure 5.03 and the corresponding token-topic wavelet entropy gradient is given in Figure 5.04.

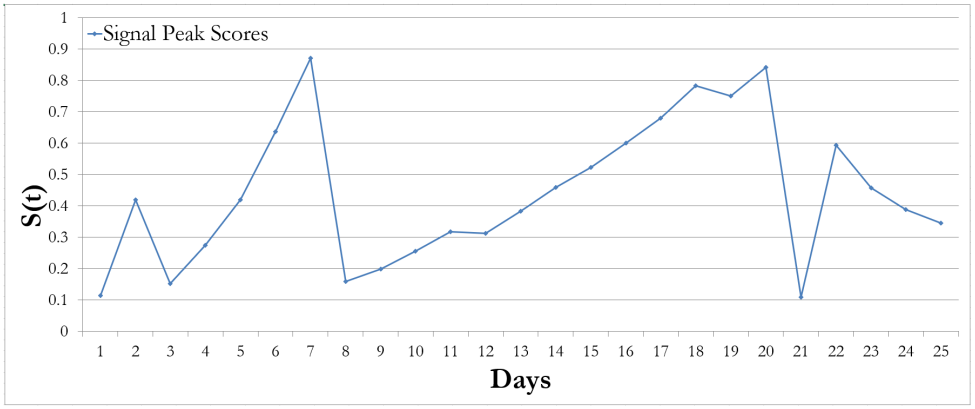


Figure 5.03. Peak Event Signal Scores. This diagram shows a typical plot of how scores determing eventful topics are identified as "short intense bursts" of signal power

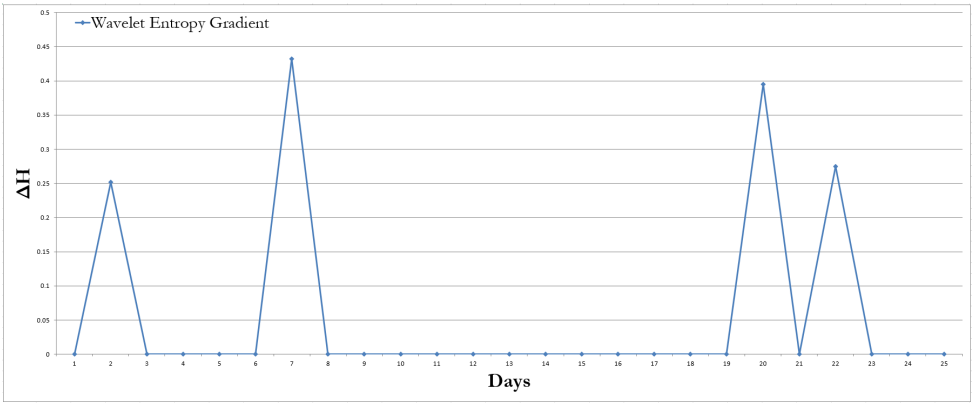


Figure 5.04. Change of Wavelet Entropy. This diagram shows how a typical signal power entropies over time as a H-measure in time. Thereby indicating if the topic wavelet is either an eventful outcome or not.

5.3.3 The Hybrid RFT Fractal Architecture

The RFT architecture which we have used in our study was developed in Section 4.6. Once the events have been identified through wavelet transformations, twitter is then queried again with the corresponding topic key

words. The retrieved tweets are then sentilyzed for the corresponding inputs in the twitter stream. A three stage FNN is built to predict the occurrence of the queried event. In the first stage, relational turbulence features like Intensity, Interference and Uncertainty are learnt over the occurrences of past events. Then, in the second stage output from each first stage FNN are fed into a Gated Recurrent Unit (GRU) structure where cell states of the Long Short Term Memory (LSTM) are constantly updated with merged forget and input gates [254]. Finally, in the third stage of the FNN architecture, the GRU cell states containing long term memory structure of peak turbulence features at the hidden layers are retrieved and act as concatenations to turbulence inputs of streaming social transactions about an interesting event.

From the uniquely discovered markovian neural network, a Convolutional Recurrent Network (CRN) [255] fractal is adopted in this study as a baseline structure to learn the fractal sub-network from pre-existing posterior confabulations. The first stage DNN architecture can be described mathematically from (4.20) to (4.27)

The second stage RFT architecture involves taking outputs from the first stage FNN framework and remembering them as cells to a larger LSTM structure. Given a hidden layer h_{t-1} from a learned FNN architecture, we wish to remember the output activations and weights of the confabulations at the peak of the episodic social turbulence attributed to the occurrence of a past event. The design of the LSTM structure is built with three gated functions that allow pass-through or blocking of convolutions from both episodic confabulations and current inputs which act as updates to the cell committing these confabulations to long term memory.

This first sigmoid gate resets old information in favor of new information that is learned from the eventful social transaction stream. The new information here corresponds to higher peaks in turbulence features learned from the first stage RFT. Mathematically, this can be written as:

$$s_t = \sigma(W_s.[h_{t-1}, n_t]) \tag{5.23}$$

Where r_t is the output of the reset gate (volatile memory), which is driven by the single layer neural network weights W_s on the convolutions of both LSTM cell inputs (h_{t-1}) and external confabulations (n_t).

The next gate is the update gate which acts as both input and forget gates of traditional LSTM models. The update gate is driven mathematically as:

$$f_t = \sigma(W_f.[h_{t-1}, n_t]) \tag{5.24}$$

Where f_t is the output of the update gate (persistent memory), which is driven by the single layer neural weights W_f on convolutions of past LSTM cell memories and current inputs. This gate decides which new information is important to add and what old information is unimportant and gets thrown away.

Finally, the last gate is the output gate. This gate decides what the next hidden state should be from memories stored in the LSTM cell. It is given mathematically as:

$$h_t = (1 - f_t) * h_{t-1} + f_t * \tilde{h_t} \tag{5.25}$$

Where,

$$\tilde{h_t} = tanh(W.[S_t * h_{t-1}, n_t]) \tag{5.26}$$

Here, $\tilde{h_t}$ represents the updated cell memory.

A detailed structure of our GRU implementation is given in Figure 5.06. Finally, in the third stage of the RFT architecture, hidden confabulation states remembered by the GRU are then passed onto the last FNN architecture which is built on a single layer convolutional perceptron fractal. The confabulation prediction outputs of the social transactions during episodic events from the GRU are concatenated to the confabulations from inputs of the model from a current social stream of transaction data. Mathematically, this is represented as:

$$H_t = A(W_t.(h_t \oplus h_{t-1})) + B \tag{5.27}$$

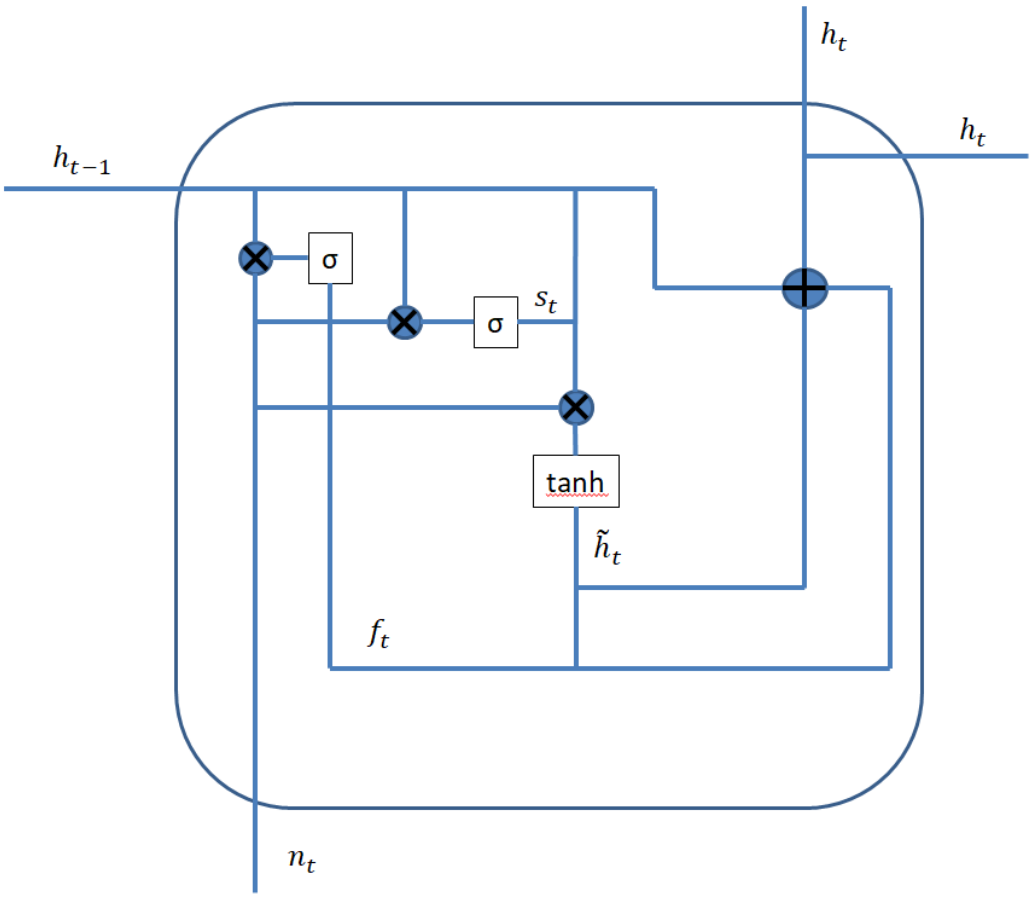


Figure 5.06. A GRU Baseline Implementation. This diagram shows the logical structure of the model's GRU implementation that remembers key relational turbulence profiles learned from past examples to be used for event prediction.

Where H_t is the next hidden layer activity, A is the activation function, W_t are the hidden layer weights, $h_t \oplus h_{t-1}$ is the concatanate of both episodic and current hidden layer activities and B is the prediction bias. This is given mathematically as:

$$B = \begin{cases} \frac{h_t - h_{t-1}}{h_{t-1}} \\ 0 \end{cases}$$

This bias is translated as gradient changes across the hidden confabulation layers.

The forward pass and loss function discovery of our model used in this chapter's study is described in Section 4.7, the Backpropagation and Fine Tuning is given in Section 4.8 and the activation and anti-alising is given in section

4.9.

5.4 Experiments and Results

The experiments were conducted on three datasets using three different algorithms. The datasets are: Twitter, Google Feed and Live Journal entries. The Twitter4J contains APIs (http://help.sentiment140.com/api) for classifying raw tweets that allows us to integrate their classifiers into our deep learning model. Their plug-in module allows us to stream tweets continuously over a span of time and their filters allowed us to query tweets by geo-locality so that we were able to detect interesting evolving events. In addition to the sentiment results obtained from their model, we cross validated the output against googles NLP API (https://cloud.google.com/natural-language/) to replicate the most accurate sentiment scores and magnitudes of context spaces and mentions.

From our model, Relational Turbulence was calculated from conditional posteriors of γ_{rl}, ϑ_{rl} and φ_{rl} as the mathematical relation of:

$$P(\tau_{rl}) = \sum_{i=1}^{n} \frac{P(\gamma_i \vee \vartheta_i)P(\vartheta_i \vee \varphi_i)P(\varphi_i \vee_i)}{N_i P(\gamma_i)P(\vartheta_i)P(\varphi_i)} \tag{5.28}$$

The inputs were tested across the RFT dynamically stacked Fractal Neural Network (FNN) and hybrid Probabilistic-Markovian (PR) based future event prediction architecture models and the learning results were compared using both F1 score and K-fold cross validation to measure both accuracy and performance of the prediction. The results are shown in the tables 1 - 4 and Figures 5.07 - 5.36.

The F1-score test was conducted on the results obtained from the experiments.

Importantly, the F1 score measures both contributions of precision and recall as important metrics to access the performance of the RFT prediction to the baseline Hybrid Probabilistic Markovian (PM) prediction of future events.

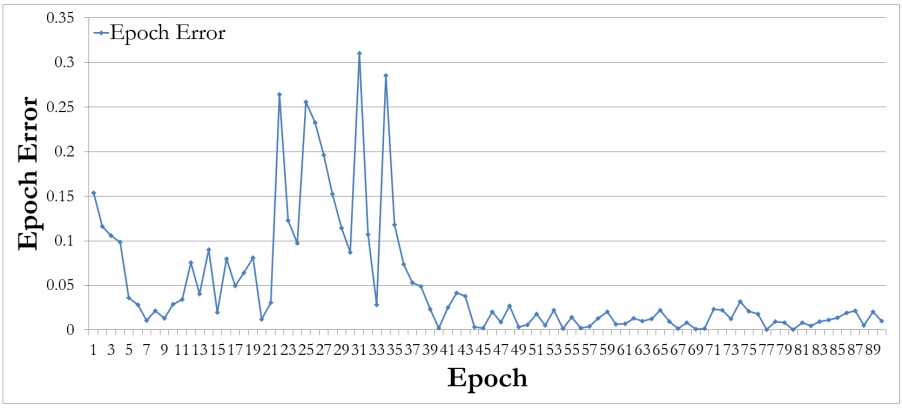


Figure 5.07. Twitter "One Belt One Road" Epoch Error

Specifically, the F1 score is given as:

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{5.29}$$

Where

$$Precision = \frac{(TruePositives)}{(TruePositives + FalsePositives)} \tag{5.30}$$

And

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{5.31}$$

The F1 score results are given in Tables 5.01 to 5.03:

Finally, during the experimentation, the full datasets obtained from the different sources (twitter, google and live-journal) were partitioned into k-subsamples. One of the subsamples was retained as the validation set for each run and the validation set was chosen in a round robin fashion for subsequent experimentation runs. A noteworthy point of mention is that K fold cross validation is used in our experimentation design to obtain a good estimate of the prediction generalization. This testing technique does not scale well to measurements of model precision. How accurately a learning model is able to predict an expected output is based on the F1-scores. K-fold validation was performed over all data streaming sources learnt and predicted

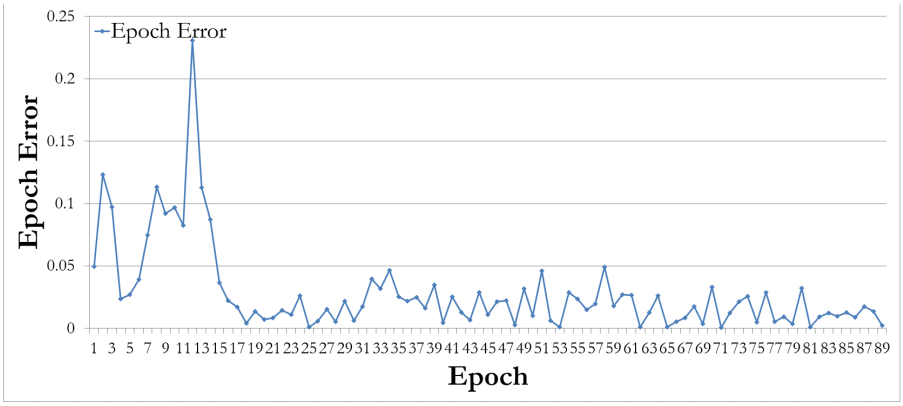


Figure 5.08. Twitter "Terrorist Attack" Epoch Error

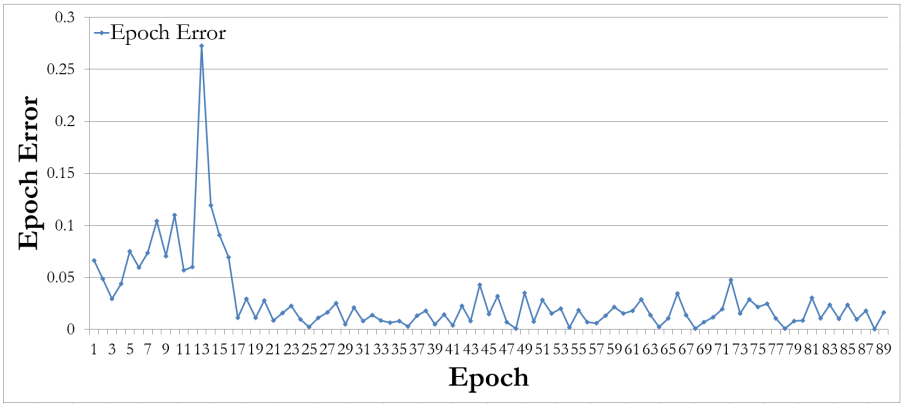


Figure 5.09. Twitter "Trade Tariff Cuts" Epoch Error

by the RFT framework across the Mean Absolute Percentage Error (MAPE) measurement of each run. Mathematically, MAPE can be expressed as:

$$\delta_{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{E_i(x) - Y_i(t)}{E_i(x)}\right| \tag{5.32}$$

Where $E_i(x)$ is the expectation at the output at data input set i and $Y_i(t)$ is the corresponding prediction over N total subsamples.

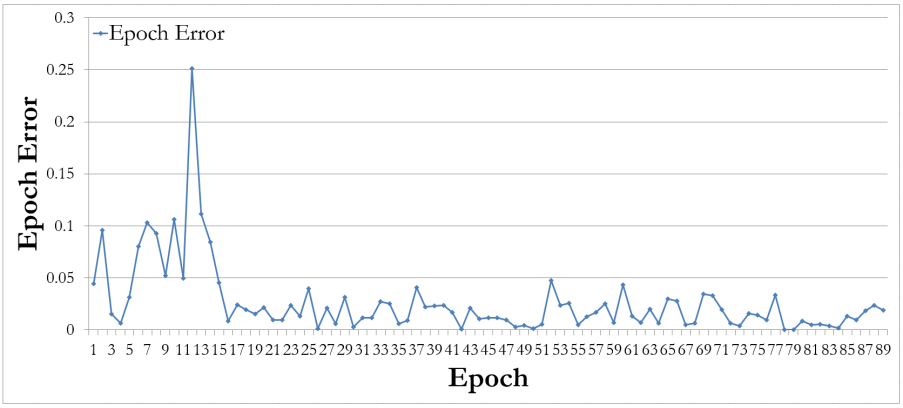


Figure 5.10. Twitter "Mexico Border" Epoch Error

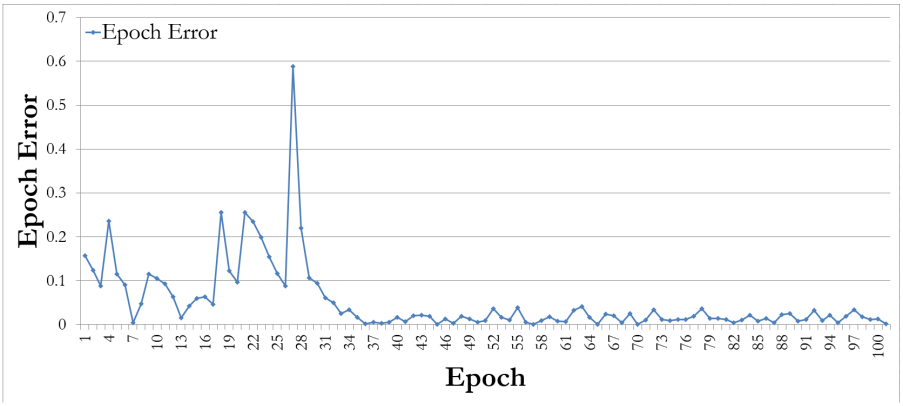


Figure 5.11. Twitter "Pacific Hurricane" Epoch Error

The tabulation of the K-fold cross validation used in our experimentation is given in Table 5.04.

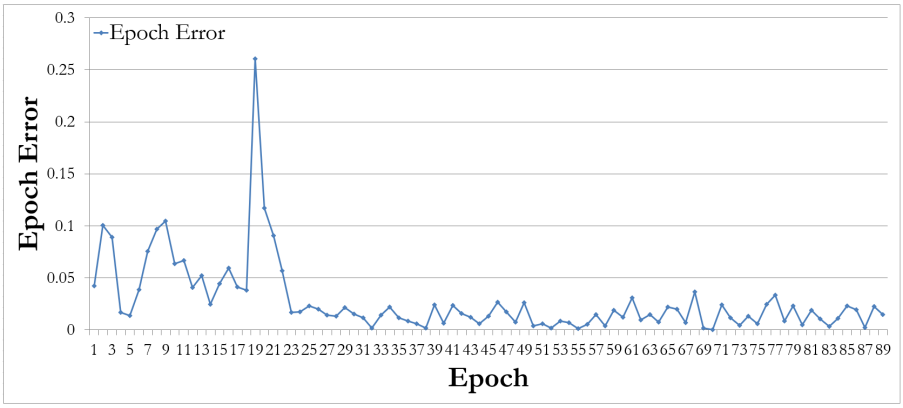


Figure 5.12. LiveJournal "One Belt One Road" Epoch Error

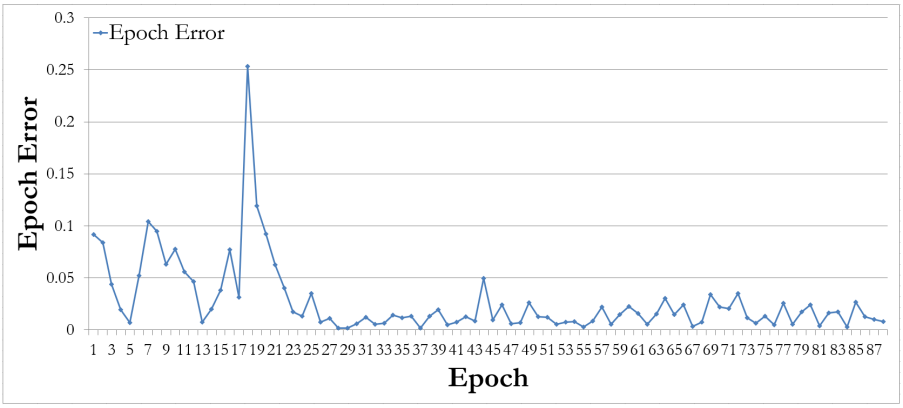


Figure 5.13. LiveJournal "Terrorist Attack" Epoch Error

5.5 Analysis and Discussions

As can be seen from the graphs, our RFT model measures comparably well to the Hybrid PM event prediction baseline model for future event occurrences. Additionally, across all sources of information streams, it is capable of measuring high F1 scores over events which have been detected by our framework from a mixed set of detected topics.

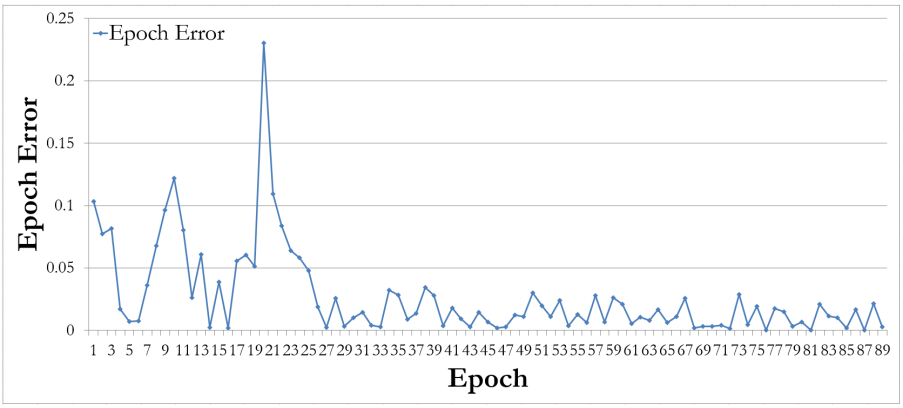


Figure 5.14. LiveJournal "Trade Tariff Cuts" Epoch Error

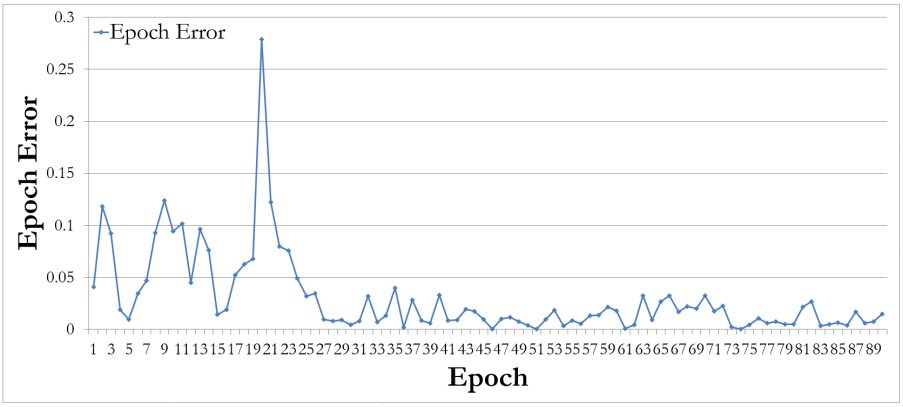


Figure 5.15. LiveJournal "Mexico Border" Epoch Error

As can be readily observed from Tables 5.01 to 5.03, prediction over events like terrorist attack and mexico border do not fare as well as other events like one belt one road and trade tariff cuts or pacific hurricane. This can intuitively be attributed to the fact that terrorist attacks and mexico border topics and their associated events have either not occurred or that have a high degree of uncertainty in their occurances. As such, positive examples for such past occurrences have been sparse and difficult to acquire and train our model adequately with.

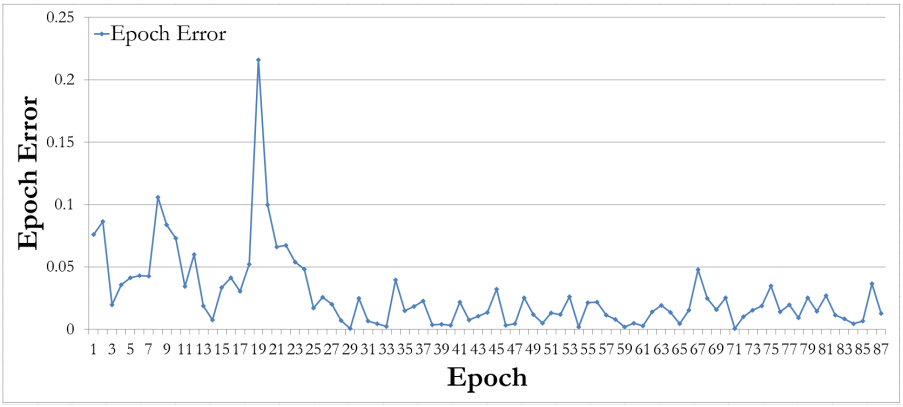


Figure 5.16. LiveJournal "Pacific Hurricane" Epoch Error

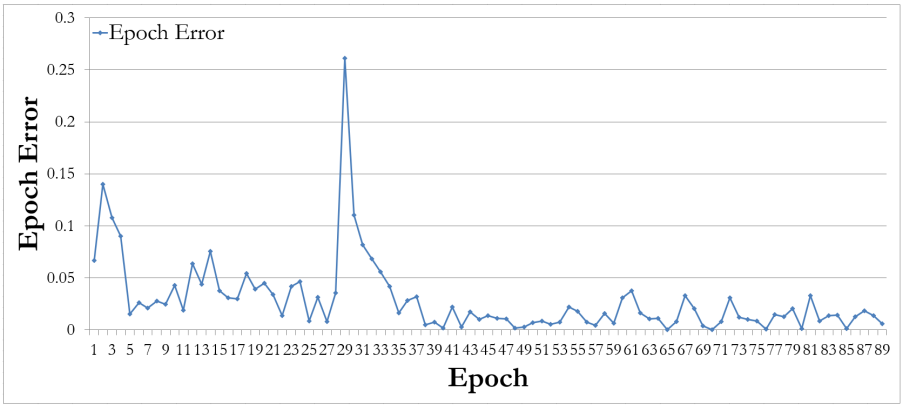


Figure 5.17. GoogleFeed "One Belt One Road" Epoch Error

Generally however, it can be observed that from Table 5.04, as the number of sub-sample windows increases over the dataset, the MAPE over all data sources decreases considerably. This means that a longer continuous training sample set will produce more accurate results from the total bag size of samples. Thus, it can be intuitively inferred that the longer the time frame spent on learning a continuous stream of social exchanges, the more accurate the prediction of events will be for topics which are currently tracked.

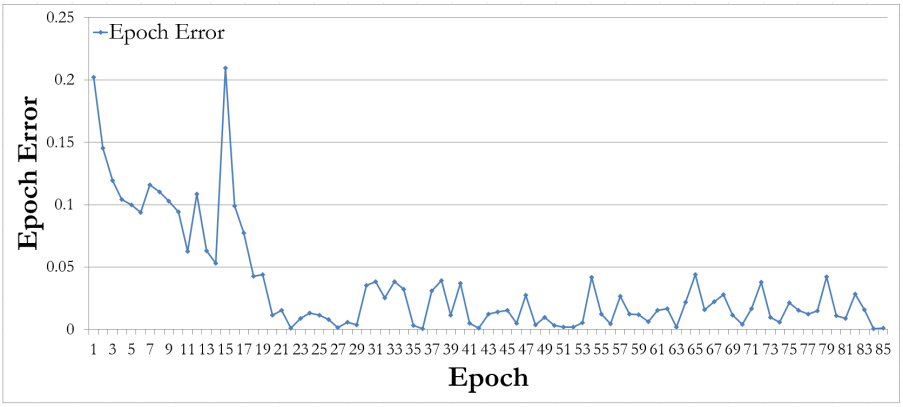


Figure 5.18. GoogleFeed "Terrorist Attack" Epoch Error

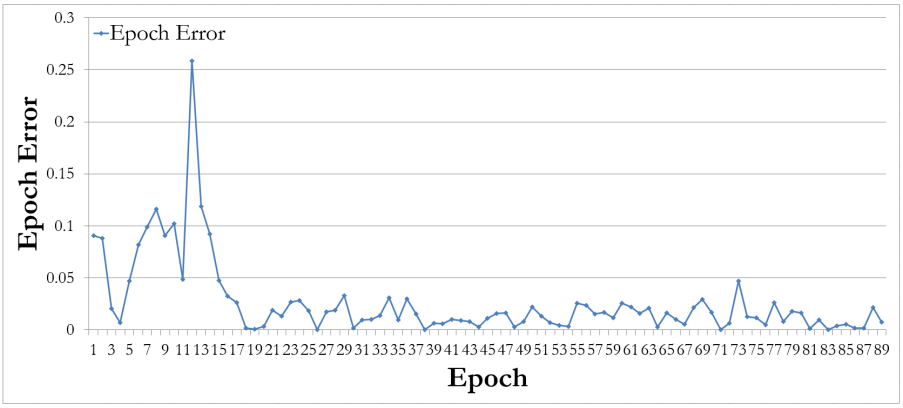


Figure 5.19. GoogleFeed "Trade Tariff Cuts" Epoch Error

5.6 Conclusions

In conclusion, our research provides new insights into event prediction from a relational intelligence perspective that could provide more accurate predictions over time. In addition, we have also developed the novel FNN framework to accomodate the complexities in anytime sequenced data. Our results show that the FNN model is capable of learning adaptively to the complexity of information received in real-time. We have demonstrated how both rela-

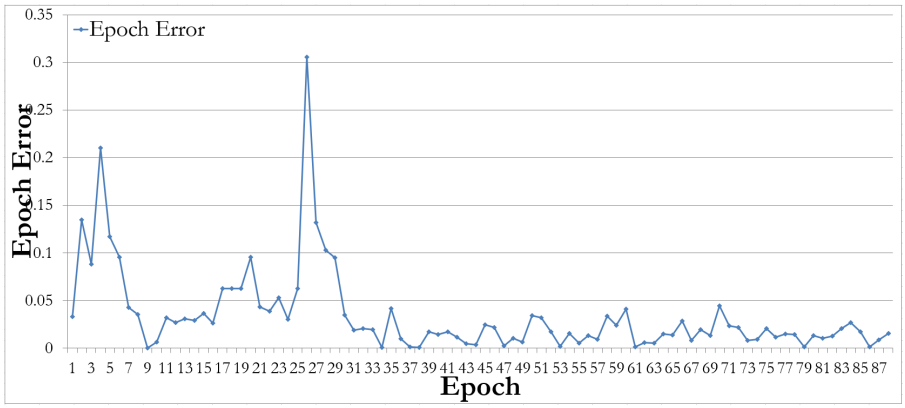


Figure 5.20. GoogleFeed "Mexico Border" Epoch Error

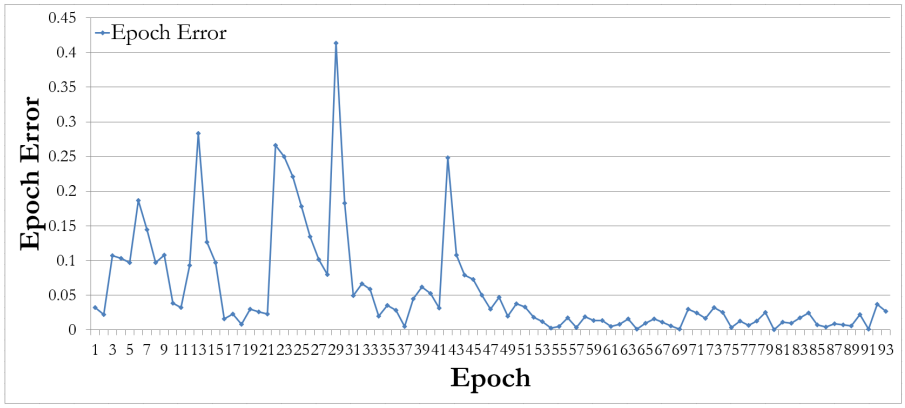


Figure 5.21. GoogleFeed "Pacific Hurricane" Epoch Error

tional turbulence and fractal intelligence can be successfully implemented in the context of three major large scale networks: Twitter, GoogleFeed and LiveJournal. Importantly, as seen from the results, this approach performed comparatively better and more efficiently to the industry standard Hybrid Probabilistic Markovian approach. Progress has been made in Event Prediction from approaches covering various perspectives. However, several key important questions still remain. All of them converge to the representational accuracy of dynamic and evolving social structure nestled within an

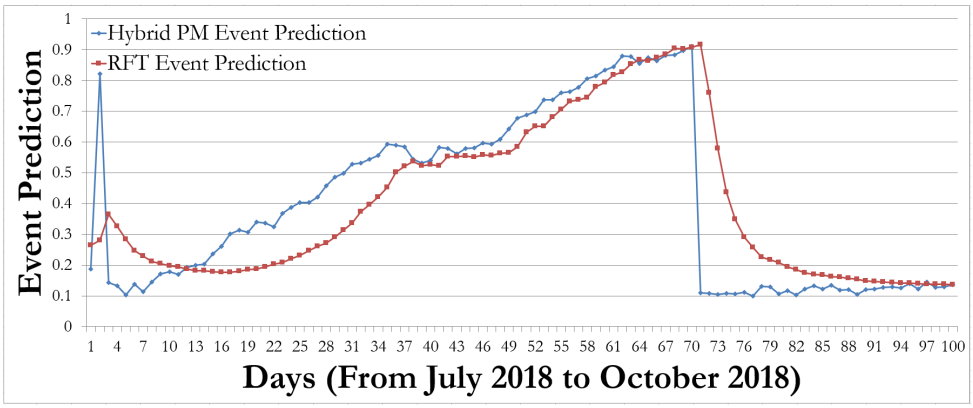


Figure 5.22. Twitter "One Belt One Road" Event Prediction

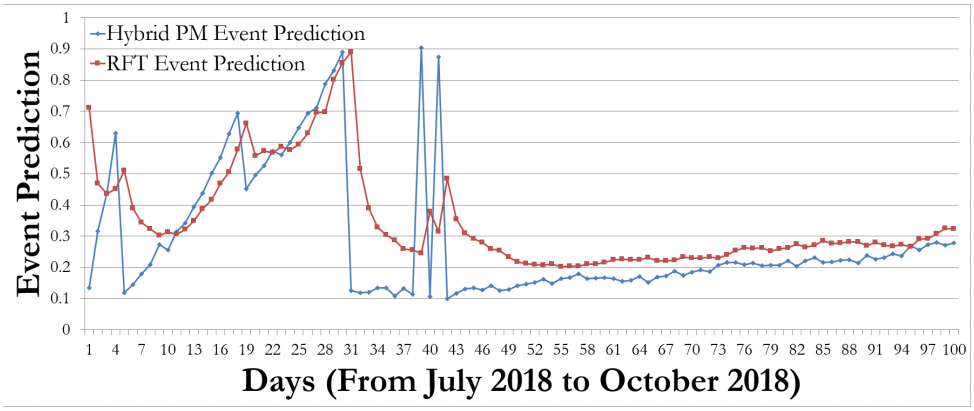


Figure 5.23. Twitter "Terrorist Attack" Event Prediction

environment of constant social shocks. Our study uncovers three pivotal long-term objectives from a relational perspective. Firstly, relational features can be used to strengthen medical, cyber security and social applications where the constant challenges between detection, recommendation, prediction, data utility and privacy are being continually addressed. Secondly, in fintech applications, relational predicates (e.g. turbulence) are determinants to market movements - closely modeled after a system of constant shocks. Thirdly, in artificial intelligence applications like computer cognition and robotics, learn-

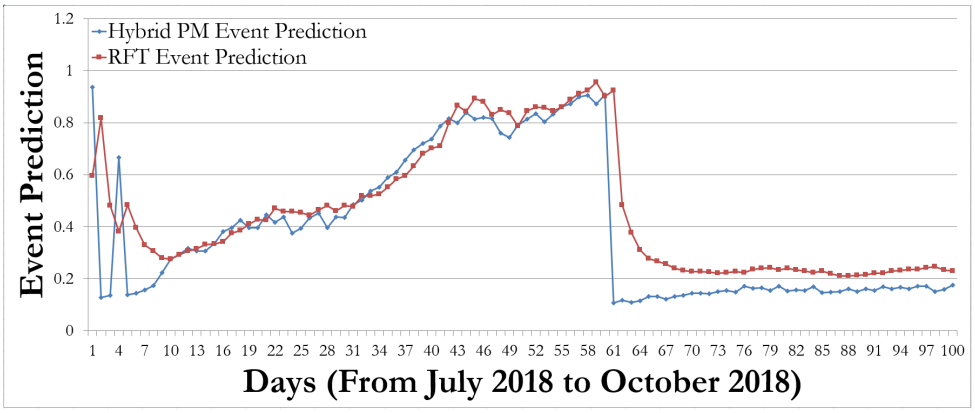


Figure 5.24. Twitter "Trade Tariff Cuts" Event Prediction

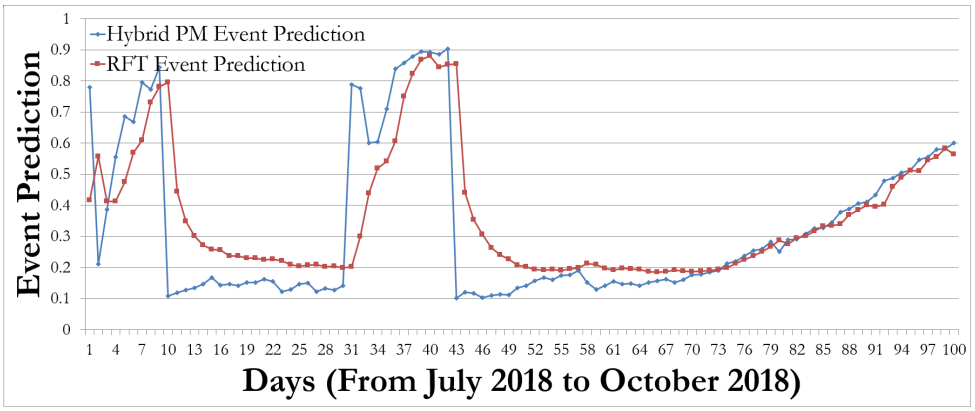


Figure 5.25. Twitter "Mexico Border" Event Prediction

ing relational features between social actors enables machines to recognize and evolve. Deep learning relational graph models appear to have considerable potential, especially in the fast growing area of social networks.

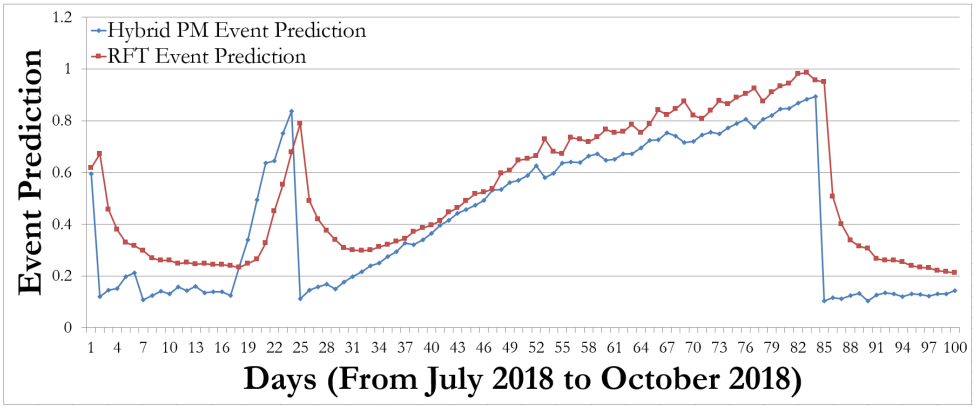


Figure 5.26. Twitter "Pacific Hurricane" Event Prediction

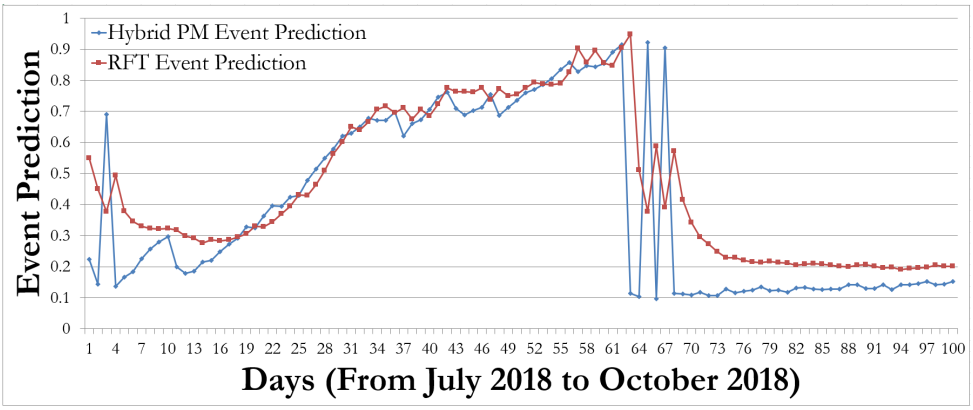


Figure 5.27. LiveJournal "One Belt One Road" Event Prediction

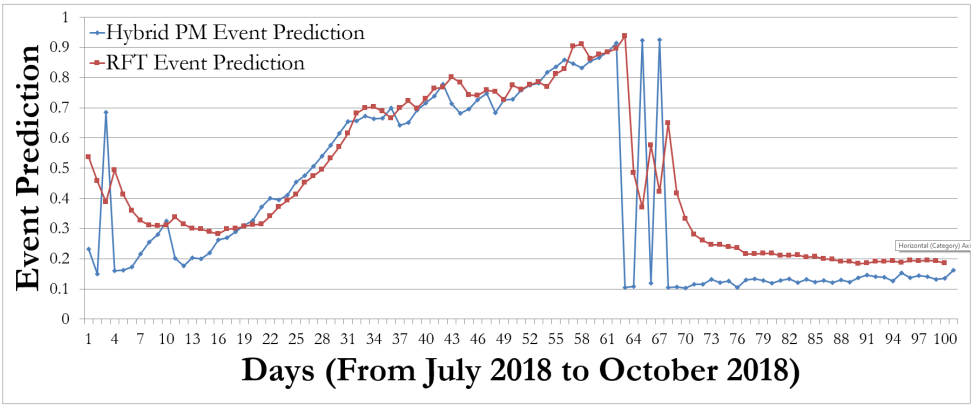


Figure 5.28. LiveJournal "Terrorist Attack" Event Prediction

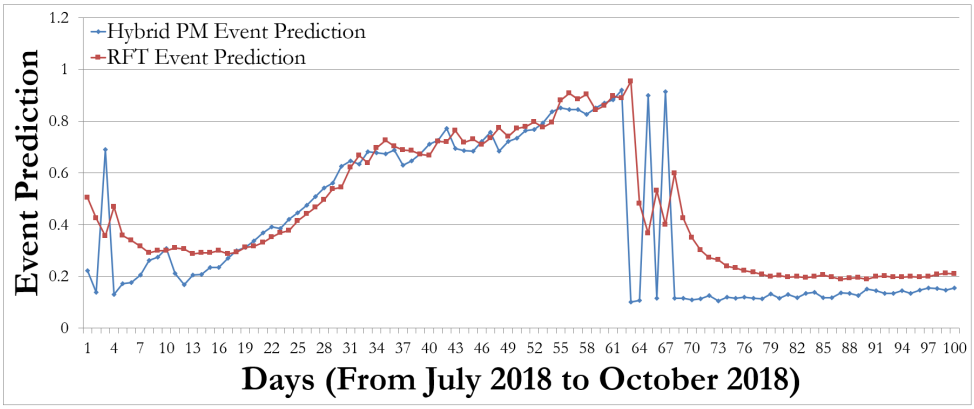


Figure 5.29. LiveJournal "Trade Tariff Cuts" Event Prediction

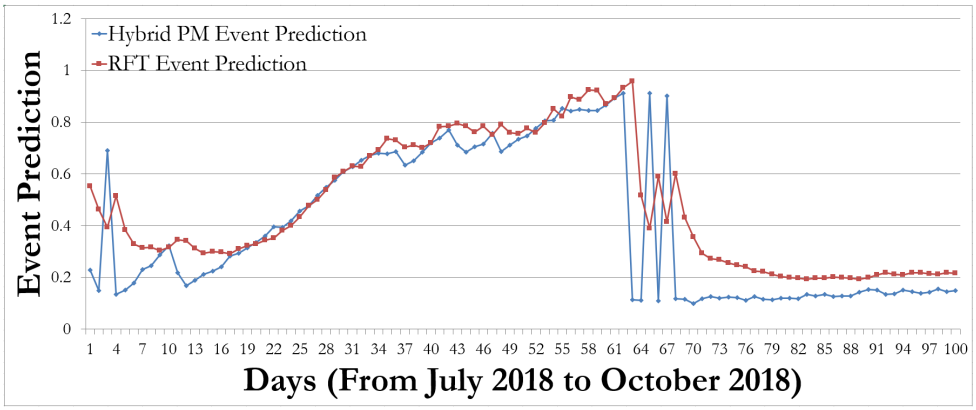


Figure 5.30. LiveJournal "Mexico Border" Event Prediction

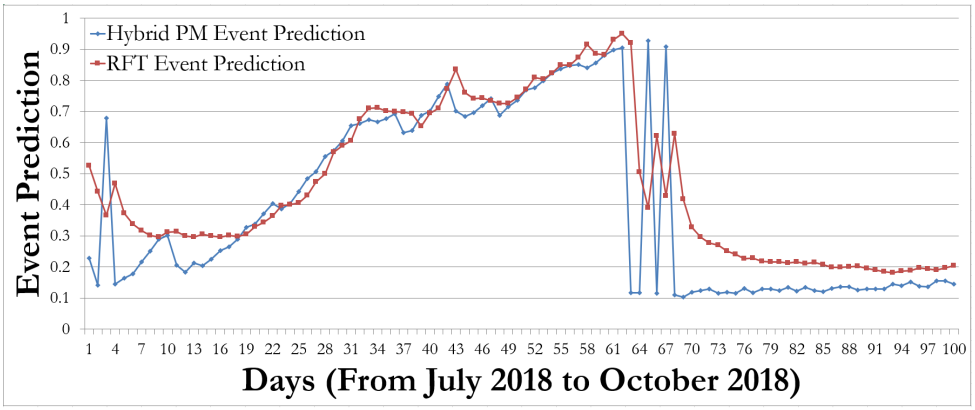


Figure 5.31. LiveJournal "Pacific Hurricane" Event Prediction

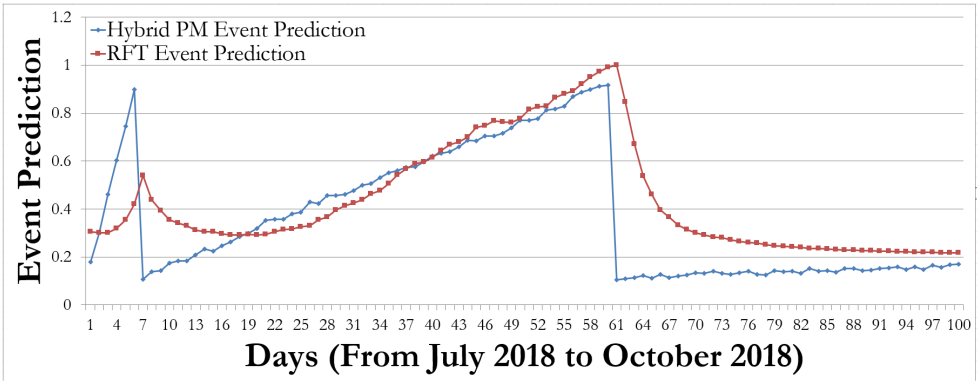


Figure 5.32. GoogleFeed "One Belt One Road" Event Prediction

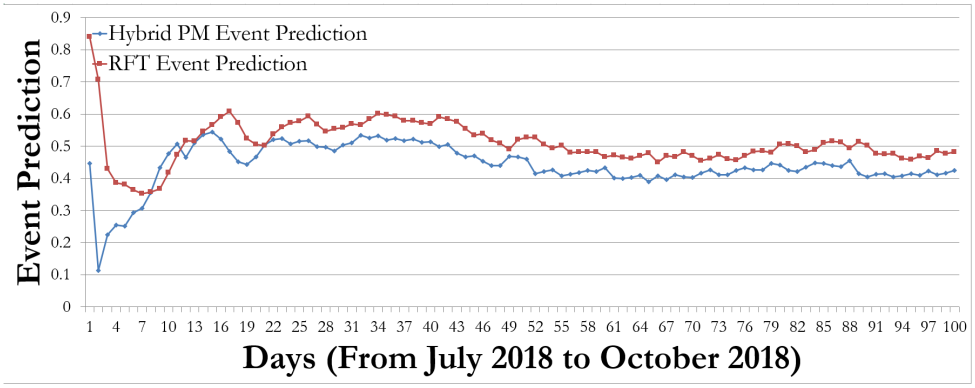


Figure 5.33. GoogleFeed "Terrorist Attack" Event Prediction

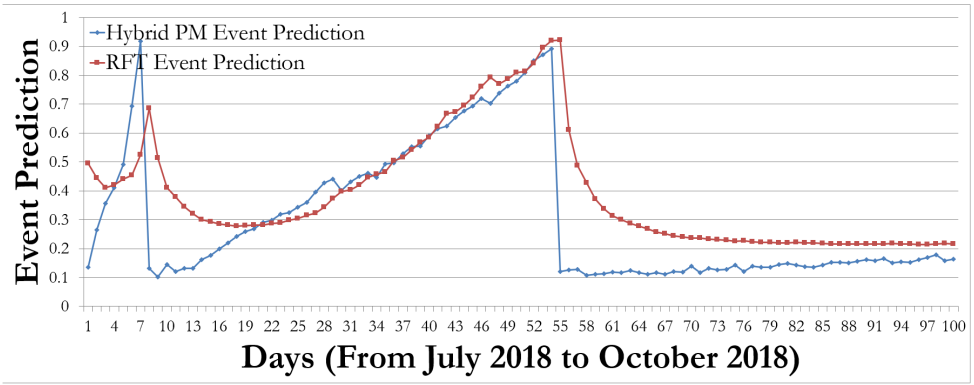


Figure 5.34. GoogleFeed "Trade Tariff Cuts" Event Prediction

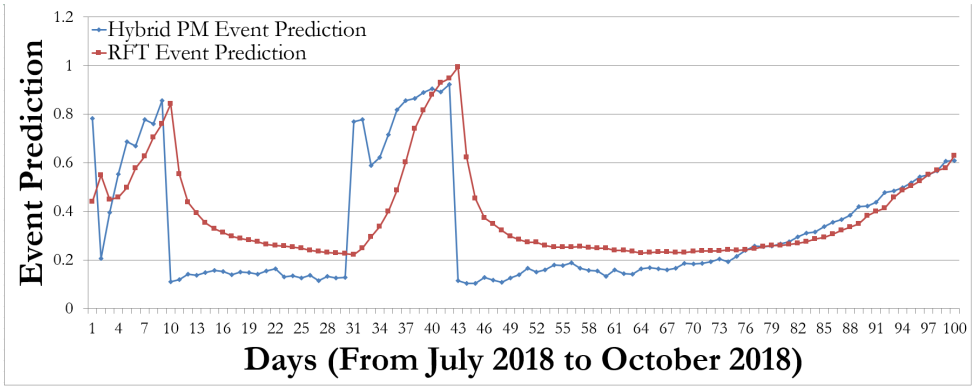


Figure 5.35. GoogleFeed "Mexico Border" Event Prediction

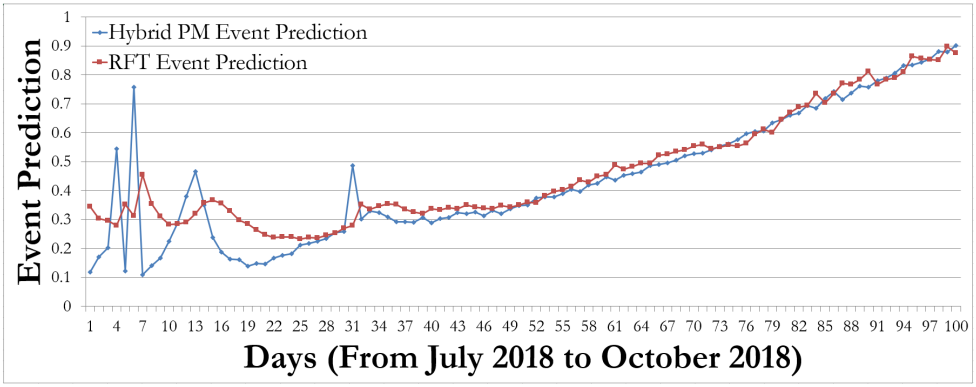


Figure 5.36. GoogleFeed "Pacific Hurricane" Event Prediction

Twitter Events F1-Score				
	Confusion Matrix		Precision / Recall	F1 - Score
One Belt One Road	TP = 16	TN = 79	0.889	0.845
	FP = 2	FN = 3	0.842	
Terrorist Attack	TP = 2	TN = 92	0.5	0.4
	FP = 2	FN = 4	0.333	
Trade Tariff Cuts	TP = 21	TN = 75	0.913	0.913
	FP = 2	FN = 2	0.913	
Mexico Border	TP = 8	TN = 84	0.8	0.667
	FP = 2	FN = 6	0.571	
Pacific Hurricane	TP = 20	TN = 66	0.625	0.740
	FP = 12	FN = 2	0.909	

Table 5.01. Table of F1 score between Hybrid PM and RFT event prediction for the Twitter validation dataset averaged over k cross validations

LiveJournal Events F1-Score				
	Confusion Matrix		Precision / Recall	F1 - Score
One Belt One Road	TP = 20	TN = 70	0.740	0.8
	FP = 7	FN = 3	0.870	
Terrorist Attack	TP = 20	TN = 72	0.769	0.833
	FP = 6	FN = 2	0.909	
Trade Tariff Cuts	TP = 18	TN = 72	0.72	0.783
	FP = 7	FN = 3	0.857	
Mexico Border	TP = 22	TN = 69	0.786	0.830
	FP = 6	FN = 3	0.88	
Pacific Hurricane	TP = 19	TN = 71	0.730	0.792
	FP = 7	FN = 3	0.864	

Table 5.02. Table of F1 score between Hybrid PM and RFT event prediction for the LiveJournal validation dataset averaged over k cross validations

GoogleFeed Events F1-Score				
	Confusion Matrix		Precision / Recall	F1 - Score
One Belt One Road	TP = 15	TN = 79	0.789	0.833
	FP = 4	FN = 2	0.882	
Terrorist Attack	TP = 22	TN = 47	0.423	0.587
	FP = 30	FN = 1	0.957	
Trade Tariff Cuts	TP = 9	TN = 88	0.818	0.857
	FP = 2	FN = 1	0.913	
Mexico Border	TP = 7	TN = 84	0.778	0.609
	FP = 2	FN = 7	0.5	
Pacific Hurricane	TP = 16	TN = 82	0.941	0.941
	FP = 1	FN = 1	0.941	

Table 5.03. Table of F1 score between Hybrid PM and RFT event prediction for the GoogleFeed validation dataset averaged over k cross validations

K	δ_{MAPE} - (SLP)	δ_{MAPE} - (DCN)	δ_{MAPE} - (RFT)
20	0.461	0.189	0.127
30	0.424	0.175	0.131
50	0.420	0.173	0.112
80	0.418	0.169	0.110
100	0.421	0.166	0.107

Table 5.04. Table of K-fold cross validated MAPE for all three learning models

CHAPTER 6

CONCLUSION

In conclusion, this research provides a wide range of breath and depth discovery into relational turbulence and event prediction. Firstly, we performed a detailed study into the techniques, approaches and methods surrounding relational pattern recognition of social network models and graphs. The extensive overview of heterogeneous information architectures and machine learning techniques which are used to uncover latent knowledge and features provides a firm and solid foundation for the development of our approach. Secondly, our study progressed towards describing and representation of relational states of heterogeneous Online Social Networks. As our first step, our methods evolved from detecting relational stability in OSNs using the MVVA technique. In our second step, we developed the RFT from the principles of relational turbulence and the Fractal Neural Network. In this approach, we offer an alternative to current Active Online Learning methodologies. Finally, in our last step, we developed a generalized event prediction model from our RFT architectures. From our results, it can be seen that our methods fare much better than current benchmarks on the scale of predicting events without sequence information. To surmize, our developments offer a wider, more generic approach to identify, detect and predict the occurance of events from OSNs.

In Chapter 2, the main contributions of the literature study include firstly, the development of a general framework model for recognizing affective and sentimental relational patterns probabilistic as states in OSNs over current state-of-the-art surveyed methods in research. Secondly, this chapters study explictly represents new knowledge of latent relational patterns within transactions of OSNs and information networks to augment the tasks of tackling real-life challenges like privacy and security. Thirdly, this chapter identifies the main problems associated with recognition based tasks like prediction,

detection, recommendation, ranking, etc. and future trends and directions are highlighted to tackle the needs and problems faced in OSNs and SISs.

In Chapter 3, the main contributions of the relational stability study include firstly, bridging the gap between temporality and stability of links in OSNs and handles dynamic link features efficiently in link prediction tasks. Secondly, a novel Hamiltonian Monte Carlo module was included as an extension to the MVVA model for scalability to big data sets. Thirdly, our experimental data shows good correlations to ground truth distributions of stable links within a Facebook clique with good accuracy performance.

In Chapter 4, an in depth study was done on relational turbulence to model relational features of OSNs between social actors. The key contributions of this chapter include firstly, developing a novel RFT model to capture key relational features used for detecting and profiling relational state transitions of eventful occurances. Secondly, the design of a novel FNN approach to adaptively learn from real-time online streaming data to identify key turbulent relationships within a given OSN. Thirdly, this chapter conducts rigorious studies on key social datasets from Twitter, Google and Enron emails. The test results show very good correlation of detected relational turbulent states to ground truths from the RTM and RFT is the clear winner from K-fold cross validation results conducted across the MAPE measurements.

In Chapter 5, the key application task of predicting general events from OSNs was further developed from the RTM and RTT perspective. Essentially, the main contributions of this chapter include firstly, using the novel RFT model developed in Chapter 4 to rigorously train relational fractals within a given topic-event context. Secondly, using the FNN architecture to adaptively learn key relational fractal structures discovered from tracking topics. Thirdly, extending the novel RFT model to include an adversarial model for robust prediction mechanisms to handle topic drifts in an MMM social transaction corpus. Finally, this chapter conducts rigorous studies on key social datasets from Twitter, GoogleFeed and LiveJournal for general event prediction tasks. The results, which compared F1 scores and K-fold cross validation measures both accuracy and performance of the prediction reveals that the novel RFT-FNN model developed is the clear winner across

predicting general events of different topics.

CHAPTER 7

FUTURE DIRECTIONS

Event prediction is a complex and evolving task which spans across wide ranging practical applications of interest. Of which, there have been five key applications that are reliant on predictive approaches to establish results of their models. A summary of each application and the corresponding contribution of this study are given below:

7.1 Online Recommender Systems

Event prediction of Online Recommendation systems are an evolving topic of interest. Many recommendation approaches and models rely on apriori data in order to make predictions about future consumer behavior in online platforms. Almost all such approaches rely on consumer data and how they interact with other entities in a publicly structured knowledge graph. The challenges with predicting events in this scenario is that firstly, relational features are not adequately represented. Only entity attributes within such knowledge graphs are updated at specified intervals. As a result, predicting events based purely on entity attributes at static time instances means that both accuracy and performance of the recommendation task is compromised. The key challenge with knowledge graphs are that they are too cumbersome to adapt to real-time anytime online data streams.

7.2 Privacy and Security Systems

Event prediction is used extensively in Privacy and Security Systems to predict hacker behavior and unauthorized access of smart data stores. Cybersecurity is a key topic of interest in the digital world today, with at least

7 in 10 protected data sources suffering intrusions on a daily basis. As sophisticated as today's security systems are, they are ill-adapted to handle anonymous requests over the internet. Adversarial attacks provide a powerful channel for hackers and malicious actors to gain access into unauthorized data stores and repositories. The key dilemma which organizations struggle with in security is the issue of sacrificing privacy over utility. In this field, predicting events based on relational features between hackers of a social network, is capable of automating complex reasoning and self-learning by massively scaling up to data. The contribution is significant in ways that can improve cyber-resilience.

7.3 Medical Information and Tele-Medicine Systems

In Medical information and tele-medicine systems, event prediction is an indispensible technology for patient monitoring systems. In large medical institutions, healthcare professionals have limited resources and attention spans to effectively diagnose a patients health status at any instance in time. Hence, they rely heavily on AI techniques to predict events like organ failure, disease outbreak, drug-patient pair matching, etc. Of recent advancements, one can look at how computer vision has evolved in its infusion into key medical devices like X-rays and MRI machines. Promising contributions and discovery of AI and ML approaches within this field show how natural language processing (NLP), deep learning and block-chain provisions for drug prescription and safety, early stage cancer cell discovery, protein systhesis identification and matching, disease matching and prevention and smart exchange of mobile health data. However, key problems faced by event prediction techniques in this industry are also due to slow data updates and over-reliance on static medical knowledge graphs. Our study contributes to this industry by predicting events based on relational features from knowledge structures which are more accurate and reliable for emergency response procedures.

7.4 Fintech and Business Intelligence

Business Intelligence has always been the cornerstone from where key decisions are made which directly impacts functional aspects of the economy. Today, companies and corporations are overwhelmed by the sheer volume of data generated by their customers from online tools through the internet. AI and ML event prediction tecnologies are already showing signs of significant growth behind the iron curtains of webpages and online portals. Traditional standalone software like spreadsheets and dashboards have been replaced by AI powered automated models that explore data, discover knowledge and process recommendations on the fly.

7.5 Education

Education is an industry which is slow evolving, cumbersome and change intolerant. The basic concepts of the pedagogical model, the domain model and the learner model have remained unchanged across the generations. Artificial Intelligence in Education (AIEd) is a vast but poorly nourished interdisciplinary field of research that investigates higher meta-forms of self-evolving learning architectures like Artificial Curiosity (AC) and Power Play (PP). Key questions remain unanswered over how AI event prediction approaches and tasks can be used to augment human capabilities today. One such field of application is in education where several cognitively challenging areas exist (e.g. learner achievement gaps, expertise development, retention, substitution, etc.).

REFERENCES

[1] G. Amodeo, R. Blanco, and U. Brefeld, "Hybrid models for future event prediction," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 1981–1984.

[2] G. M. Weiss and H. Hirsh, "Event prediction: learning from ambiguous examples," in *Working Notes of the NIPS9298 Workshop on Learning from Ambiguous and Complex Examples*, 1998.

[3] Y. Yang, T. Pierce, and J. G. Carbonell, "A study on retrospective and on-line event detection," *ISPRS International Journal of Geo-Information*, vol. 6, no. 3, p. 88, 1998.

[4] M. Cordeiro and J. Gama, "Online social networks event detection: a survey," in *Solving Large Scale Learning Tasks. Challenges and Algorithms*. Springer, 2016, pp. 1–41.

[5] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard, "Multiscale event detection in social media," *Data Mining and Knowledge Discovery*, vol. 29, no. 5, pp. 1374–1405, 2015.

[6] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[7] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, "Stability of graph communities across time scales," *Proceedings of the National Academy of Sciences*, vol. 107, no. 29, pp. 12 755–12 760, 2010.

[8] J. Zhang, X. Tao, L. Tan, J. C.-W. Lin, H. Li, and L. Chang, "On link stability detection for online social networks," in *International Conference on Database and Expert Systems Applications*. Springer, 2018, pp. 320–335.

[9] D. H. Solomon, L. K. Knobloch, J. A. Theiss, and R. M. McLaren, "Relational turbulence theory: Explaining variation in subjective experiences and communication within romantic relationships," *Human Communication Research*, vol. 42, no. 4, pp. 507–532, 2016.

[10] C. C. Aggarwal, Y. Xie, and P. S. Yu, "Towards community detection in locally heterogeneous networks," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 391–402.

[11] L. Backstrom and J. Leskovec, "Link prediction in social networks using computationally efficient topological features," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 73–80.

[12] D. Haunani Solomon, *Relational Turbulence Model*, 06 2015.

[13] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger, "Social media analytics–challenges in topic discovery, data collection, and data preparation," *International journal of information management*, vol. 39, pp. 156–168, 2018.

[14] D. Knights, M. C. Mozer, and N. Nicolov, "Detecting topic drift with compound topic models," in *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[15] J. A. Theiss and D. H. Solomon, "A relational turbulence model of communication about irritations in romantic relationships," *Communication Research*, vol. 33, no. 5, pp. 391–418, 2006. [Online]. Available: https://doi.org/10.1177/0093650206291482

[16] T. A. Snijders, "Markov chain monte carlo estimation of exponential random graph models," *Journal of Social Structure*, vol. 3, no. 2, pp. 1–40, 2002.

[17] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.

[18] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[19] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *arXiv preprint arXiv:1802.02871*, 2018.

[20] D. H. Solomon and L. K. Knobloch, "Relationship uncertainty, partner interference, and intimacy within dating relationships," *Journal of Social and Personal Relationships*, vol. 18, no. 6, pp. 804–820, 2001. [Online]. Available: https://doi.org/10.1177/0265407501186004

[21] L. K. Knobloch and J. A. Theiss, "Relational turbulence theory applied to the transition from deployment to reintegration," *Journal of Family Theory & Review*, vol. 10, no. 3, pp. 535–549, 2018.

[22] F. Gao, K. Musial, C. Cooper, and S. Tsoka, "Link prediction methods and their accuracy for different social networks and network metrics," *Scientific Programming*, vol. 2015, p. 1, 2015.

[23] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.

[24] C. Rudin, B. Letham, A. Salleb-Aouissi, E. Kogan, and D. Madigan, "Sequential event prediction with association rules," in *Proceedings of the 24th annual conference on learning theory*, 2011, pp. 615–634.

[25] M. Granroth-Wilding and S. Clark, "What happens next? event prediction using a compositional neural network model," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[26] K.-I. Goh, G. Salvi, B. Kahng, and D. Kim, "Skeleton and fractal scaling in complex networks," *Physical review letters*, vol. 96, no. 1, p. 018701, 2006.

[27] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.

[28] S. Petrovic, "Real-time event detection in massive streams," *Data Mining and Knowledge Discovery*, vol. 29, no. 5, pp. 1374–1405, 2013.

[29] H.-O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and fractals: new frontiers of science*. Springer Science & Business Media, 2006.

[30] J. Beniger, *The control revolution: Technological and economic origins of the information society*. Harvard university press, 2009.

[31] J. Zhang, L. Tan, and X. Tao, "On relational learning and discovery in social networks: a survey," *International Journal of Machine Learning and Cybernetics*, pp. 1–18, 2018.

[32] K. S. K. Chung, M. Piraveenan, and L. Hossain, "Topology of online social networks," *Encyclopedia of Social Network Analysis and Mining*, pp. 2191–2202, 2014.

[33] H. Lune and B. L. Berg, *Qualitative research methods for the social sciences*. Pearson Higher Ed, 2016.

[34] C. W.-k. Leung, E.-P. Lim, D. Lo, and J. Weng, "Mining interesting link formation rules in social networks," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 209–218.

[35] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 635–644.

[36] F. Qiao, P. Li, X. Zhang, Z. Ding, J. Cheng, and H. Wang, "Predicting social unrest events with hidden markov models using gdelt," *Discrete Dynamics in Nature and Society*, vol. 2017, 2017.

[37] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[38] R. Stademann and K. Gehlhaus, "Process and routing system for dynamic traffic control in a communication network," July 27 1999, uS Patent 5,930,249.

[39] G. Hardin, "Method and apparatus for network bandwidth allocation," Sep. 23 2014, uS Patent 8,843,978.

[40] M. Christopher, *Logistics & supply chain management*. Pearson UK, 2016.

[41] M. Levi, "Assessing the trends, scale and nature of economic cybercrimes: overview and issues," *Crime, Law and Social Change*, vol. 67, no. 1, pp. 3–20, 2017.

[42] V. K. Gandhi and T. N.-S. I. Thanjavur, "An overview study on cyber crimes in internet," *Journal of Information Engineering and Applications*, vol. 2, no. 1, pp. 1–5, 2012.

[43] W. Hu, H. Wang, Z. Qiu, C. Nie, L. Yan, and B. Du, "An event detection method for social networks based on hybrid link prediction and quantum swarm intelligence," *World Wide Web*, vol. 20, no. 4, pp. 775–795, 2017.

[44] K. Radinsky and E. Horvitz, "Mining the web to predict future events," in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 255–264.

[45] F. Jin, W. Wang, P. Chakraborty, N. Self, F. Chen, and N. Ramakrishnan, "Tracking multiple social media for stock market event prediction," in *Industrial Conference on Data Mining*. Springer, 2017, pp. 16–30.

[46] M. Bojanic, M. Gnjatovic, M. Secujski, and V. Delic, "Application of dimensional emotion model in automatic emotional speech recognition," in *Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on*. IEEE, 2013, pp. 353–356.

[47] D. Gatica-Perez, "Automatic nonverbal analysis of social interaction in small groups: A review," *Image and vision computing*, vol. 27, no. 12, pp. 1775–1787, 2009.

[48] H. Gunes and M. Pantic, "Automatic, dimensional and continuous emotion recognition," *International Journal of Synthetic Emotions (IJSE)*, vol. 1, no. 1, pp. 68–99, 2010.

[49] L. Getoor and C. P. Diehl, "Link mining: a survey," *SIGKDD Explorations*, vol. 7, pp. 3–12, 2005.

[50] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[51] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009, pp. 322–335.

[52] X. Feng, J. Zhao, and K. Xu, "Link prediction in complex networks: a clustering perspective," *The European Physical Journal B*, vol. 85, no. 1, p. 3, 2012.

[53] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of relational structure," in *ICML*, vol. 1, 2001, pp. 170–177.

[54] D. Schall, "Link prediction in directed social networks," *Social Network Analysis and Mining*, vol. 4, no. 1, p. 157, 2014.

[55] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 243–252.

[56] O. J. Mengshoel, R. Desai, A. Chen, and B. Tran, "Will we connect again? machine learning for link prediction in mobile social networks," in *Eleventh Workshop on Mining and Learning with Graphs*, 2013.

[57] L. Munasinghe and R. Ichise, "Time score: A new feature for link prediction in social networks," *IEICE TRANSACTIONS on Information and Systems*, vol. 95, no. 3, pp. 821–828, 2012.

[58] L. Munasinghe and R. Ichise, "Time aware index for link prediction in social networks." in *DaWaK*. Springer, 2011, pp. 342–353.

[59] L. Munasinghe and R. Ichise, "Link prediction in social networks using information flow via active links," *IEICE TRANSACTIONS on Information and Systems*, vol. 96, no. 7, pp. 1495–1502, 2013.

[60] L. Munasinghe and R. Ichise, "Multi-class link prediction in social networks," in *proceedings of 27th Annual Conference of the Japanese Society for Artificial Intelligence*, 2013.

[61] F. Esposito, S. Ferilli, T. M. Basile, and N. Di Mauro, "Social networks and statistical relational learning: a survey," *International Journal of Social Network Mining*, vol. 1, no. 2, pp. 185–208, 2012.

[62] B. Pang, L. Lee et al., "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[63] M. V. Redmond, "Uncertainty reduction theory," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2015.

[64] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, p. 8, 2009.

[65] R. Bunescu and R. J. Mooney, "Relational markov networks for collective information extraction," in *ICML-2004 Workshop on Statistical Relational Learning*, 2004.

[66] Y. Z. M. W. S. C. H. M. T. Zhao, Peilin and J. Huang, "Adaptive cost-sensitive online classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 214–228, 2019.

[67] S. Mohamad, "Active learning for data streams." Ph.D. dissertation, Bournemouth University, 2017.

[68] A. Weiler, M. Grossniklaus, and M. H. Scholl, "Run-time and task-based performance of event detection techniques for twitter," in *International Conference on Advanced Information Systems Engineering*. Springer, 2015, pp. 35–49.

[69] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1354–1361.

[70] V. Ramanathan, B. Yao, and L. Fei-Fei, "Social role discovery in human events," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2475–2482.

[71] J. Read, L. Martino, and D. Luengo, "Efficient monte carlo methods for multi-dimensional learning with classifier chains," *Pattern Recognition*, vol. 47, no. 3, pp. 1535–1546, 2014.

[72] G. Nightingale, N. J. Boogert, K. N. Laland, and W. Hoppitt, "Quantifying diffusion in social networks: a bayesian approach," *Animal social networks*, pp. 38–52, 2015.

[73] J. Wang, P. Zhao, and S. C. Hoi, "Cost-sensitive online classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2425–2438, 2014.

[74] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction." in *AAAI*, vol. 8, 2008, pp. 677–682.

[75] S. Sengupta and Y. Chen, "Spectral clustering in heterogeneous networks," *Statistica Sinica*, pp. 1081–1106, 2015.

[76] H. Wang and D.-Y. Yeung, "Towards bayesian deep learning: A survey," *arXiv preprint arXiv:1604.01662*, 2016.

[77] E. Mossel, A. Sly, and O. Tamuz, "Asymptotic learning on bayesian social networks," *Probability Theory and Related Fields*, vol. 158, no. 1-2, pp. 127–157, 2014.

[78] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar, "Bayesian learning in social networks," *The Review of Economic Studies*, vol. 78, no. 4, pp. 1201–1236, 2011.

[79] A. Rodriguez, "Modeling the dynamics of social networks using bayesian hierarchical blockmodels," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 3, pp. 218–234, 2012.

[80] B. Taskar, E. Segal, and D. Koller, "Probabilistic classification and clustering in relational data," in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001, pp. 870–878.

[81] Y. Rao, H. Xie, J. Li, F. Jin, F. L. Wang, and Q. Li, "Social emotion classification of short text via topic-level maximum entropy model," *Information & Management*, vol. 53, no. 8, pp. 978–986, 2016.

[82] Y. Ding, C. Liu, P. Zhao, and S. C. H. Hoi, "Large scale kernel methods for online auc maximization," *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 91–100, 2017.

[83] J. Lu, P. Zhao, and S. C. Hoi, "Online passive-aggressive active learning," *Machine Learning*, vol. 103, no. 2, pp. 141–183, 2016.

[84] D. J. Hand, "Measuring classifier performance: a coherent alternative to the area under the roc curve," *Machine learning*, vol. 77, no. 1, pp. 103–123, 2009.

[85] S. Ravindran and G. Aghila, "A data-independent reusable projection (dirp) technique for dimension reduction in big data classification using k-nearest neighbor (k-nn)," *National Academy Science Letters*, pp. 1–9, 2019.

[86] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 96–103.

[87] P. Skryjomski, B. Krawczyk, and A. Cano, "Speeding up k-nearest neighbors classifier for large-scale multi-label learning on gpus," *Neurocomputing*, vol. 354, pp. 10–19, 2019.

[88] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[89] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific reports*, vol. 2, 2012.

[90] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *IEEE transactions on knowledge and data engineering*, vol. 20, no. 2, pp. 172–188, 2008.

[91] S. Zhong, "Efficient online spherical k-means clustering," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 5. IEEE, 2005, pp. 3180–3185.

[92] A. Dubey, A. Hefny, S. Williamson, and E. P. Xing, "A nonparametric mixture model for topic modeling over time," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 530–538.

[93] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, "Detecting communities and their evolutions in dynamic social networksa bayesian approach," *Machine learning*, vol. 82, no. 2, pp. 157–189, 2011.

[94] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 677–685.

[95] D. He, D. Liu, D. Jin, and W. Zhang, "A stochastic model for detecting heterogeneous link communities in complex networks." in *AAAI*, 2015, pp. 130–136.

[96] F. Yang and F. Zhang, "Community detection for multilayer heterogeneous network," *arXiv preprint arXiv:1705.05967*, 2017.

[97] J. Tang, X. Wang, and H. Liu, "Integrating social media data for community detection," in *Modeling and Mining Ubiquitous Social Media*. Springer, 2012, pp. 1–20.

[98] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 119–128.

[99] K. Sheshadri, C.-W. Hang, and M. Singh, "Framing matters: Predicting framing changes and legislation from topic news patterns," *arXiv preprint arXiv:1802.05762*, 2018.

[100] M. Durr, M. Werner, and M. Maier, "Re-socializing online social networks," in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010, pp. 786–791.

[101] V. Sadovykh, D. Sundaram, and S. Piramuthu, "Do online social networks support decision-making?" *Decision support systems*, vol. 70, pp. 15–30, 2015.

[102] A. Farasat, A. Nikolaev, S. N. Srihari, and R. H. Blair, "Probabilistic graphical models in modern social network analysis," *Social Network Analysis and Mining*, vol. 5, no. 1, p. 62, 2015.

[103] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM workshop on Online social networks*. ACM, 2009, pp. 37–42.

[104] D. R. Hunter, P. N. Krivitsky, and M. Schweinberger, "Computational statistical methods for social network models," *Journal of Computational and Graphical Statistics*, vol. 21, no. 4, pp. 856–882, 2012.

[105] Y. Fan and C. R. Shelton, "Learning continuous-time social network dynamics," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 161–168.

[106] Z. Li, Z. Pan, Y. Zhang, G. Li, and G. Hu, "Efficient community detection in heterogeneous social networks," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[107] P. De Meo, A. Nocera, G. Quattrone, and D. Ursino, "A conceptual framework for community detection, characterisation and membership in a social internetworking scenario," *International Journal of Data Mining, Modelling and Management*, vol. 6, no. 1, pp. 22–48, 2014.

[108] D. M. Romero and J. M. Kleinberg, "The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter." in *ICWSM*, 2010.

[109] L. Li, K. Yue, and Z. Sun, *A Probabilistic Approach for Inferring Latent Entity Associations in Textual Web Contents*, 04 2019, pp. 3–18.

[110] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *journal of the Association for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[111] D. Song and D. A. Meyer, "Link sign prediction and ranking in signed directed social networks," *Social Network Analysis and Mining*, vol. 5, no. 1, p. 52, 2015.

[112] F. Buccafurri, G. Lax, A. Nocera, and D. Ursino, "Discovering links among social networks," *Machine learning and knowledge discovery in databases*, pp. 467–482, 2012.

[113] S. Kok and P. Domingos, "Learning markov logic networks using structural motifs," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 551–558.

[114] F. Liu, B. Liu, C. Sun, M. Liu, and X. Wang, "Deep belief network-based approaches for link prediction in signed social networks," *Entropy*, vol. 17, no. 4, pp. 2140–2169, 2015.

[115] X. Liu, W. Liu, T. Murata, and K. Wakita, "A framework for community detection in heterogeneous multi-relational networks," *Advances in Complex Systems*, vol. 17, no. 06, p. 1450018, 2014.

[116] C. Deng, Z. Lv, W. Liu, J. Huang, D. Tao, and X. Gao, "Multi-view matrix decomposition: A new scheme for exploring discriminative information." in *IJCAI*, 2015, pp. 3438–3444.

[117] M. A. Ahmad, Z. Borbora, J. Srivastava, and N. Contractor, "Link prediction across multiple social networks," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 911–918.

[118] H. Xu, Y. Hu, Z. Wang, J. Ma, and W. Xiao, "Core-based dynamic community detection in mobile social networks," *Entropy*, vol. 15, no. 12, pp. 5419–5438, 2013.

[119] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations," in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 917–922.

[120] T.-A. N. Pham, X. Li, G. Cong, and Z. Zhang, "A general recommendation model for heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3140–3153, 2016.

[121] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

[122] Q. Meng, S. Tafavogh, and P. J. Kennedy, "Community detection on heterogeneous networks by multiple semantic-path clustering," in *Computational Aspects of Social Networks (CASoN), 2014 6th International Conference on*. IEEE, 2014, pp. 7–12.

[123] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, 2010, pp. 856–864.

[124] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[125] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, pp. 1–43, 2017.

[126] M. Steyvers and T. Griffiths, "Probabilistic topic models," *Handbook of latent semantic analysis*, vol. 427, no. 7, pp. 424–440, 2007.

[127] J. Li, Y. Song, Z. Wei, and K.-F. Wong, "A joint model of conversational discourse and latent topics on microblogs," *Computational Linguistics*, vol. 44, no. 4, pp. 719–754, 2018.

[128] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "Metafac: community discovery via relational hypergraph factorization," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 527–536.

[129] J. Scripps, P.-N. Tan, F. Chen, and A.-H. Esfahanian, "A matrix alignment approach for link prediction," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[130] B. Sigurleifsson, A. Anbarasu, and K. Kangur, "The count-min sketch data structure and its uses within computer science," 2019.

[131] R. Hisano, "Semi-supervised graph embedding approach to dynamic link prediction," *arXiv preprint arXiv:1610.04351*, 2016.

[132] A. Potgieter, K. A. April, R. J. Cooke, and I. O. Osunmakinde, "Temporality in link prediction: Understanding social complexity," *Emergence: Complexity and Organization*, vol. 11, no. 1, p. 69, 2009.

[133] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2015.

[134] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, 2015, pp. 2503–2511.

[135] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[136] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[137] K. Cho, T. Raiko, and A. Ilin, "Parallel tempering is efficient for learning restricted boltzmann machines," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–8.

[138] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. L. P. Chen, "Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3-d meshes," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2268–2281, 2017.

[139] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 115–124.

[140] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22 073–22 078, 2009.

[141] D. Hand and C. Anagnostopoulos, "A better beta for the h measure of classification performance preprint," *arXiv preprint arXiv:1202.2564*.

[142] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[143] N. P. Nguyen, T. N. Dinh, Y. Shen, and M. T. Thai, "Dynamic social community detection and its applications," *PloS one*, vol. 9, no. 4, p. e91431, 2014.

[144] W. Liu, T. Murata, and X. Liu, "Community detection on heterogeneous networks," *The 27th Annual Conference of the Japanese Society for Artificial Intelligence*, 2013.

[145] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, 2013, pp. 1151–1156.

[146] T. Murata and S. Moriyasu, "Link prediction of social networks based on weighted proximity measures," in *Proceedings of the IEEE/WIC/ACM international conference on web intelligence*. IEEE Computer Society, 2007, pp. 85–88.

[147] X. Zheng, S. Zhu, J. Gao, and H. Mamitsuka, "Instance-wise weighted nonnegative matrix factorization for aggregating partitions with locally reliable clusters." in *IJCAI*, 2015, pp. 4091–4097.

[148] Y. Sun, C. C. Aggarwal, and J. Han, "Relation strength-aware clustering of heterogeneous information networks with incomplete attributes," *Proceedings of the VLDB Endowment*, vol. 5, no. 5, pp. 394–405, 2012.

[149] J. Cheng, L. Li, M. Leng, W. Lu, Y. Yao, and X. Chen, "A divisive spectral method for network community detection," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 3, p. 033403, 2016.

[150] D. Zhou and C. J. Burges, "Spectral clustering and transductive learning with multiple views," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1159–1166.

[151] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition." in *AAAI*, 2014, pp. 2149–2155.

[152] Z. Tao, H. Liu, S. Li, and Y. Fu, "Robust spectral ensemble clustering," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 367–376.

[153] P. Singla and P. Domingos, "Discriminative training of markov logic networks," in *AAAI*, vol. 5, 2005, pp. 868–873.

[154] P. Domingos and M. Richardson, "1 markov logic: A unifying framework for statistical relational learning," *Statistical Relational Learning*, p. 339, 2007.

[155] W. Ching, S. Zhang, and M. Ng, "On multi-dimensional markov chain models," *Pacific Journal of Optimization*, vol. 3, no. 2, 2007.

[156] B. Taskar, P. Abbeel, M.-F. Wong, and D. Koller, "Relational markov networks," *Introduction to statistical relational learning*, pp. 175–200, 2007.

[157] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in *Proceedings of the 2008 SIAM international conference on data mining*. SIAM, 2008, pp. 822–833.

[158] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. IEEE, 2009, pp. 1016–1021.

[159] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph laplacians for semi–supervised learning," in *Advances in Neural Information Processing Systems*, 2006, pp. 67–74.

[160] L. Tang, X. Wang, and H. Liu, "Scalable learning of collective behavior," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1080–1091, 2012.

[161] F. Radicchi, "Detectability of communities in heterogeneous networks," *Physical Review E*, vol. 88, no. 1, p. 010801, 2013.

[162] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[163] L. Deng, D. Yu et al., "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[164] Y. Bengio et al., "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[165] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *arXiv preprint arXiv:1711.04043*, 2017.

[166] S. Reed, Y. Chen, T. Paine, A. v. d. Oord, S. Eslami, D. Rezende, O. Vinyals, and N. de Freitas, "Few-shot autoregressive density estimation: Towards learning to learn distributions," *arXiv preprint arXiv:1710.10304*, 2017.

[167] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *arXiv preprint arXiv:1611.05763*, 2016.

[168] E. Basegmez, "The next generation neural networks: Deep learning and spiking neural networks," in *Advanced Seminar in Technical University of Munich*, 2014, pp. 1–40.

[169] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[170] C. F. Higham and D. J. Higham, "Deep learning: An introduction for applied mathematicians," *arXiv preprint arXiv:1801.05894*, 2018.

[171] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with evolving streaming data," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 597–612.

[172] D. Sahoo, Q. Pham, J. Lu, and S. C. Hoi, "Online deep learning: Learning deep neural networks on the fly," *arXiv preprint arXiv:1711.03705*, 2017.

[173] M. Prince, "Does active learning work? a review of the research," *Journal of engineering education*, vol. 93, no. 3, pp. 223–231, 2004.

[174] X. Zhu, P. Zhang, X. Lin, and Y. Shi, "Active learning from data streams," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 757–762.

[175] J. Lu, P. Zhao, and S. Hoi, "Online sparse passive aggressive learning with kernels," pp. 675–683, 06 2016.

[176] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 27–39, 2014.

[177] J. Lafferty, A. McCallum, F. Pereira et al., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Journal Of Machine Learning Research (JMLR)*, 2001.

[178] A. Pakman and L. Paninski, "Auxiliary-variable exact hamiltonian monte carlo samplers for binary distributions," in *Advances in neural information processing systems*, 2013, pp. 2490–2498.

[179] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1633–1685, 2009.

[180] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[181] J. Z. Leibo, C. d. M. d'Autume, D. Zoran, D. Amos, C. Beattie, K. Anderson, A. G. Castañeda, M. Sanchez, S. Green, A. Gruslys et al., "Psychlab: A psychology laboratory for deep reinforcement learning agents," *arXiv preprint arXiv:1801.08116*, 2018.

[182] W. Liu and L. Lü, "Link prediction based on local random walk," *EPL (Europhysics Letters)*, vol. 89, no. 5, p. 58007, 2010.

[183] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International journal of machine learning and cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[184] Q.-s. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

[185] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, vol. 5, 2015.

[186] S. Vishwanathan, K. M. Borgwardt, and N. N. Schraudolph, "Fast computation of graph kernels," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*. MIT Press, 2006, pp. 1449–1456.

[187] G.-B. Huang, Y.-Q. Chen, and H. A. Babri, "Classification ability of single hidden layer feedforward neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 799–801, 2000.

[188] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning shared structures from multiple tasks," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 137–144.

[189] H.-K. Peng, J. Zhu, D. Piao, R. Yan, and Y. Zhang, "Retweet modeling using conditional random fields," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 336–343.

[190] H. Neuvirth, M. Ozery-Flato, J. Hu, J. Laserson, M. S. Kohn, S. Ebadollahi, and M. Rosen-Zvi, "Toward personalized care management of patients at risk: the diabetes case study," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 395–403.

[191] L. De Vine, G. Zuccon, B. Koopman, L. Sitbon, and P. Bruza, "Medical semantic similarity with a neural language model," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. ACM, 2014, pp. 1819–1822.

[192] J. R. A. Moniz and D. Krueger, "Nested lstms," *arXiv preprint arXiv:1801.10308*, 2018.

[193] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun, "Multi-layer representation learning for medical concepts," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1495–1504.

[194] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2133–2136.

[195] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[196] B. Hutchinson, L. Deng, and D. Yu, "A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4805–4808.

[197] Q. Chen and R. Wu, "Cnn is all you need," *arXiv preprint arXiv:1712.09662*, 2017.

[198] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[199] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[200] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning et al., "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," *arXiv preprint arXiv:1802.01561*, 2018.

[201] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Problems*, vol. 34, no. 1, p. 014004, 2017.

[202] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, P. J. Liu, X. Liu, M. Sun, P. Sundberg, H. Yee et al., "Scalable and accurate deep learning for electronic health records," *arXiv preprint arXiv:1801.07860*, 2018.

[203] S. Peng, R. You, H. Wang, C. Zhai, H. Mamitsuka, and S. Zhu, "Deepmesh: deep semantic representation for improving large-scale mesh indexing," *Bioinformatics*, vol. 32, no. 12, pp. i70–i79, 2016.

[204] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[205] X. Tao, X. Zhou, J. Zhang, and J. Yong, "Sentiment analysis for depression detection on social networks," in *International Conference on Advanced Data Mining and Applications*. Springer, 2016, pp. 807–810.

[206] X. Zhou, X. Tao, M. M. Rahman, and J. Zhang, "Coupling topic modelling in opinion mining for social media analysis," in *Proceedings of the International Conference on Web Intelligence*. ACM, 2017, pp. 533–540.

[207] J. Zhang, L. Tan, X. Tao, X. Zheng, Y. Luo, and J. C.-W. Lin, "Slind: Identifying stable links in online social networks," in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 813–816.

[208] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[209] N. P. Nguyen, M. A. Alim, T. N. Dinh, and M. T. Thai, "A method to detect communities with stability in social networks," *Social Network Analysis and Mining*, vol. 4, no. 1, pp. 1–15, 2014.

[210] T. Chakraborty, S. Srinivasan, N. Ganguly, S. Bhowmick, and A. Mukherjee, "Constant communities in complex networks," *arXiv preprint arXiv:1302.5794*, 2013.

[211] L. Getoor and C. P. Diehl, "Link mining: a survey," *Acm Sigkdd Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[212] A. Özcan and Ş. G. Öğüdücü, "Multivariate temporal link prediction in evolving social networks," in *Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on*. IEEE, 2015, pp. 185–190.

[213] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[214] C. J. Needham, J. R. Bradford, A. J. Bulpitt, and D. R. Westhead, "A primer on learning in bayesian networks for computational biology," *PLoS computational biology*, vol. 3, no. 8, p. e129, 2007.

[215] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.

[216] J. Sohl-Dickstein, M. Mudigonda, and M. R. DeWeese, "Hamiltonian monte carlo without detailed balance," *arXiv preprint arXiv:1409.5191*, 2014.

[217] M. Girolami, B. Calderhead, and S. A. Chin, "Riemannian manifold hamiltonian monte carlo," *arXiv preprint arXiv:0907.1100*, 2009.

[218] H. Meyer, H. Simma, R. Sommer, M. Della Morte, O. Witzel, U. Wolff, A. Collaboration et al., "Exploring the hmc trajectory-length dependence of autocorrelation times in lattice qcd," *Computer physics communications*, vol. 176, no. 2, pp. 91–97, 2007.

[219] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[220] R. Cohen and S. Havlin, *Complex networks: structure, robustness and function*. Cambridge university press, 2010.

[221] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[222] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.

[223] J. Zhang and S. Y. Philip, *Broad Learning Through Fusions*. Springer, 2019.

[224] C. Gardella, O. Marre, and T. Mora, "A tractable method for describing complex couplings between neurons and population rate," *eneuro*, vol. 3, no. 4, pp. ENEURO–0160, 2016.

[225] Y. Li, Z.-L. Zhang, and J. Bao, "Mutual or unrequited love: Identifying stable clusters in social networks with uni-and bi-directional links," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2012, pp. 113–125.

[226] S. Lu, Z. Wei, and L. Li, "A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization," *Computational Optimization and Applications*, vol. 51, pp. 551–573, 03 2012.

[227] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[228] G. Tur, L. Deng, D. Hakkani-Tür, and X. He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5045–5048.

[229] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European Conference on Computer Vision*. Springer, 2016, pp. 646–661.

[230] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[231] D. HAUNANI SOLOMON and J. Theiss, "A longitudinal test of the relational turbulence model of romantic relationship development," *Personal Relationships*, vol. 15, pp. 339 – 357, 08 2008.

[232] W. Wilmot, J. Hocker, J. Arthur, J. W. Petty, J. J. Martocchio, H. R. Cheeseman, E. Biech, and R. J. Rosania, *Interpersonal Conflict 9th Edition*. McGraw-Hill Higher Education, 2007.

[233] R. M. McLaren, D. H. Solomon, and J. S. Priem, "The effect of relationship characteristics and relational communication on experiences of hurt from romantic partners," *Journal of Communication*, vol. 62, no. 6, pp. 950–971, 2012.

[234] J. R. Siegert and G. H. Stamp, "our first big fight as a milestone in the development of close relationships," *Communications Monographs*, vol. 61, no. 4, pp. 345–360, 1994.

[235] T. D. Afifi, "feeling caughtin stepfamilies: Managing boundary turbulence through appropriate communication privacy rules," *Journal of Social and Personal Relationships*, vol. 20, no. 6, pp. 729–755, 2003.

[236] D. H. Solomon and J. A. Samp, "Power and problem appraisal: Perceptual foundations of the chilling effect in dating relationships," *Journal of Social and Personal Relationships*, vol. 15, no. 2, pp. 191–209, 1998.

[237] C. A. Surra, "Reasons for changes in commitment: Variations by courtship type," *Journal of Social and Personal Relationships*, vol. 4, no. 1, pp. 17–33, 1987.

[238] L. A. Baxter and G. Pittman, "Communicatively remembering turning points of relational development in heterosexual romantic relationships," *Communication Reports*, vol. 14, no. 1, pp. 1–17, 2001.

[239] S. Keshmiri, H. Sumioka, J. Nakanishi, and H. Ishiguro, "Emotional state estimation using a modified gradient-based neural architecture with weighted estimates," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 4371–4378.

[240] N. P. Garg, S. Favre, H. Salamin, D. Hakkani Tür, and A. Vinciarelli, "Role recognition for meeting participants: an approach based on lexical information and social network analysis," in *Proceedings of the 16th ACM international conference on Multimedia*. ACM, 2008, pp. 693–696.

[241] L. Simeonova, "Gradient emotional analysis," 2017.

[242] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving deep neural networks," *arXiv preprint arXiv:1703.00548*, 2017.

[243] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2013.

[244] J. Gideon, S. Khorram, Z. Aldeneh, D. Dimitriadis, and E. M. Provost, "Progressive neural networks for transfer learning in emotion recognition," *arXiv preprint arXiv:1706.03256*, 2017.

[245] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[246] R. Arthur and H. T. P. Williams, "The human geography of twitter," *CoRR*, vol. abs/1807.04107, 2018.

[247] J. Weng and B.-S. Lee, "Event detection in twitter," in *Fifth international AAAI conference on weblogs and social media*, 2011.

[248] M. Cordeiro, "Twitter event detection: combining wavelet analysis and topic inference summarization," in *International Workshop on Algorithms and Models for the Web-Graph*, 2011, pp. 113–125.

[249] B. Letham, C. Rudin, and D. Madigan, "Sequential event prediction," *Machine Learning*, vol. 93, no. 2, pp. 357–380, Nov 2013. [Online]. Available: https://doi.org/10.1007/s10994-013-5356-5

[250] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 784–793.

[251] L. Du, W. Buntine, and M. Johnson, "Topic segmentation with a structured topic model," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 190–200.

[252] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine learning*, vol. 42, no. 1-2, pp. 143–175, 2001.

[253] A. Weiler, M. Grossniklaus, and M. H. Scholl, "Evaluation measures for event detection techniques on twitter data streams," in *British International Conference on Databases*. Springer, 2015, pp. 108–119.

[254] K. Cho, B. van Merrienboer, aglar Gülehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[255] L. Feng, S. Liu, and J. Yao, "Music genre classification with paralleling recurrent convolutional neural network," *arXiv preprint arXiv:1712.08370*, 2017.